Composable Q-Functions for Pedestrian Car Interactions

Christian Muench [1,2] and Dariu M. Gavrila [1]

***Abstract*— We propose a novel algorithm that predicts the interaction of pedestrians with cars within a Markov Decision Process framework. It leverages the fact that Q-functions may be composed in the maximum-entropy framework, thus the solutions of two sub-tasks may be combined to approximate the full interaction problem. Sub-task one is the interaction-free navigation of a pedestrian in an urban environment and sub-task two is the interaction with an approaching car (deceleration, waiting etc.) without accounting for the environmental context (e.g. street layout). We propose a regularization scheme motivated by the soft-Bellman-equations and illustrate its necessity. We then analyze the properties of the algorithm in detail with a toy model. We find that as long as the interaction-free sub-task is modelled well with a Q-function, we can learn a representation of the interaction between a pedestrian and a car.**

I. INTRODUCTION

Autonomous vehicles (AVs) need to be able to navigate urban environments in a time efficient manner while avoiding collisions with vulnerable road users such as pedestrians. Simply extrapolating the current velocity and direction of a pedestrian into the future to determine potential collision points is insufficient as it does not consider any environmental context (street layout) or interaction effects (may yield to approaching car). AVs that predict future trajectories of pedestrians in this manner may behave intolerably defensive and may simply freeze in crowded scenarios. Therefore, predicting the intentions of pedestrians is important so that AVs may anticipate the reaction of a pedestrian to its actions (accelerating, no change, braking) which results in more desirable trajectories that are safe, comfortable and, time efficient.

Previous work either discards the interaction of cars and pedestrians entirely [1], [2], limits the interaction to one motion type (continue or stop) while reducing the environment to a simplified representation (where is curb?) [3] or employs statistical learning approaches that are tested on datasets with limited interactions that may not generalize well to unseen environments and are untested on stop-go motion types usually encountered in the intelligent vehicle domain [4].

Regression based approaches that handle the environmental context and interaction with other agents need to deal with a multitude of combinations of obstacle configurations, street layouts and agent-agent interactions, which might limit generalizability. We want to investigate an approach that decouples the environmental context from the interactions, reducing the combinatorial explosion of *"environment configurations times interaction configurations"*. We leverage the key insight of [5] that two Q-functions in the maximum-entropy framework [6] of two distinct reinforcement learning tasks may be combined to obtain an algorithm that solves both tasks without the need of retraining or additional data collection. We propose a novel algorithm that separates the problem into two sub-tasks, one involving interaction-free movement of a pedestrian and another that describes the interaction of a pedestrian with a car. All of this is integrated into the same framework instead of relying on two separate algorithms. We find that the separation is successfully reproducing the actual interaction behaviour for a toy model using the soft-Q-learning algorithm (SQL) introduced by [7], validating the compositionality of certain robotics tasks as discussed in [5]. We propose another toy model for pedestrian-car interactions to validate our proposed algorithm.

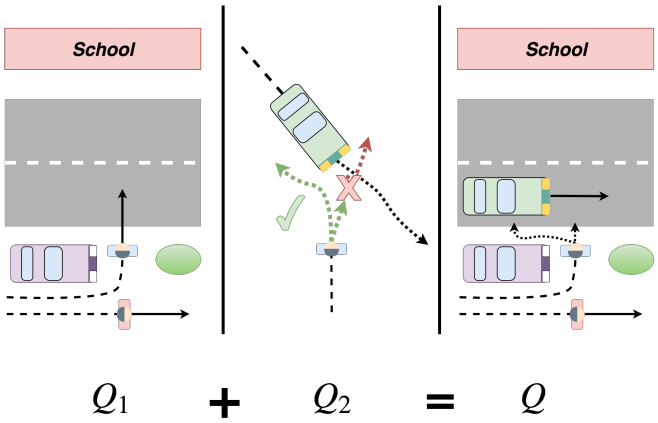


Fig. 1: (**left**) A Q-function Q_1 describing the interaction-free navigation of a pedestrian towards a school building. (**middle**) A Q-function Q_2 describing the interaction of a pedestrian and a car. (**right**) Combination of Q_1 and Q_2 results in a Q-function that describes the interaction-free navigation of a pedestrian **and** the interaction with a car (pedestrian yields).

II. RELATED WORK

A sizeable amount of work exists on the problem of forecasting the intentions of pedestrians. We will list publications that we consider closest to ours. [8], [2], [1], [9] reason about the paths that pedestrians may choose given

1 Intelligent Vehicles Group, Cognitive Robotics Department, Technical University Delft, The Netherlands; {C.Muench,D.M.Gavrila}@tudelft.nl;
2 Department of Environment Perception, Daimler AG, Ulm, Germany

context information such as the environmental semantics (street layout, obstacles, etc.) or traffic light phase [1], [9]. Another set of publications [10], [11], [12] ignore context information while considering the interactions of multiple pedestrians and predicting their collective (collision-free) paths. [13], [4], [14] and [15] combine the interactions of multiple agents and static obstacles (e.g. a wall) to obtain collision-free trajectories that may be used to navigate through obstacle rich crowded spaces with an autonomous robot. All the above publications solely focus on trajectories or environment features whereas [16] and [3] analyse the intentions of pedestrians to enter the street by including additional observations. In particular, [3] address this problem by adding pedestrian's head orientation as an intention feature to assess the criticality of situations. Other features that may be used are the human pose [17], [18] and optical flow [16] to determine the state of a pedestrian.

Inferring the future path of a pedestrian is inherently difficult given the multi-modality of possible motion patterns. In particular, constant velocity based Kalman filters fail to adjust to sudden changes in the dynamics [16]. [3] improves on these types of models by considering a switching-linear dynamical system in combination with multiple intention features that enable faster adjustments to the observed pedestrian dynamics at the curb. [1] does something similar via jump Markov processes which model a sudden change in a pedestrian's intended goals (heading direction), though stop-go types of motion were not considered. [13] considers different classes of trajectories (homotopy classes) that an agent may choose such as swerving to the left or right or choosing different goal states that are far apart. Models based on GANs and Recurrent-Neural Networks such as [11] and [4] are also able to provide multi-modal solutions. However, this does not apply to neural network based models such as [12], [14] and [15].

The combination of environmental features and the interaction of multiple agents is our focus. [4] applies a variational autoencoder to sample the candidate trajectories which are finetuned by a recurrent neural network that receives an environment map and social pooling features of the other participants. [14] and [15] apply deep reinforcement learning to learn an interaction model for a robot with humans and obstacles in crowded environments. Global path planning is handled separately and not performed by the deep RL algorithm. [13] integrates global planning and interaction of agents in a maximum-entropy framework. The underlying rewards are learned via inverse reinforcement learning and the trajectories are sampled using Hamiltonian MCMC sampling. [19] take the solution of a MDP in the maximum-entropy framework to obtain a force which is integrated into their social forces model that can describe agent-agent interactions and planning at the same time.

The decomposition of Q-functions has been discussed before in e.g. [20]. The authors propose to decompose a problem into sub-tasks, each with its own reward and corresponding Q-function and outline an algorithm that converges to the optimal Q-function. They point out that simply adding up the Q-functions may yield sub-optimal policies, though [5] made clear that this approach may still be feasible in real-world settings for the maximum-entropy framework. Additionally, [20] test their approach on a race car that drives around in a small sized grid-world while avoiding obstacles where they decomposed the sub-tasks into one for navigation and one for obstacle avoidance. Similarly, we propose composing Q-functions to separate interaction-free navigation from interactions with other agents, though to the best of our knowledge we are the first to propose this decomposition so that we may imitate the behaviour of pedestrians and cars with no predefined reward function. The approach integrates global path planning and interaction in one framework. Our contributions are as follows. 1) We validate the compositionality of Q-functions for the SQL algorithm - as proposed by [5] - for interacting agents in a toy model. 2) We propose to learn the interaction Q-function directly via gradient descent and argue that this is sensible given the domain 3) We propose a regularization scheme based on the soft-Bellman-equation so that function-approximators such as neural networks learn proper interaction soft-Q-functions. We explore our proposals on a toy model and point out future research avenues.

III. METHODOLOGY

A. Overview

Human behaviour may be described as driven by rewards that are partially hidden from others or even the person itself as they are mostly the product of the unconscious inner workings of the persons brain. We will describe the interaction behaviour of a pedestrian and a car with a "Markov Decision Process" (MDP). The MDP describes how the state s_t (e.g. position) of an agent (pedestrian, car) changes given an action a_t (e.g. pedestrian steps to the left) at time step t (time is discrete). Additionally, the agent receives a reward $r(a_t, s_t)$ when performing an action. The goal of an agent is to maximize the total reward by taking appropriate actions.

Additionally, we assume that actions are not optimal, i.e. the agent may choose an action that does not result in the maximum expected reward. A common way to describe this situation is the maximum-entropy framework [6] where the policy is expressed in terms of an energy based model $\pi \sim \exp(Energy)$

$$\pi(a_t|s_t) \sim \exp\left(\frac{1}{\alpha} Q(a_t, s_t)\right) \tag{1}$$

Where the "energy" - also known as soft-Q-function - $Q(a_t, s_t)$ is the expected future reward given that the agent performs the action a_t which can be expressed in terms of the value function $V(s_t)$

$$\begin{aligned} Q(a_t, s_t) &= r(a_t, s_t) + \sum_{s_{t+1}} p(s_{t+1}|s_t, a_t) V(s_{t+1}) \\ &= r(a_t, s_t) + V(\hat{s}_{t+1}(s_t, a_t)) \end{aligned} \tag{2}$$

We will only consider deterministic environment transitions where $p(s_{t+1}|s_t,a_t)$ reduces to a delta-function $\delta(\hat{s}_{t+1}(s_t,a_t)-s_{t+1})$ thereby eliminating the summation over possible future states. The value function can be expressed in terms of the Q-function

$$V(s_t) = \alpha \log \int \exp(\frac{1}{\alpha} Q(a_t,s_t))da_t \tag{3}$$

Equations 2 and 3 correspond to the soft-Bellman-equations for V and Q. α is a "temperature" parameter that determines the degree of optimality of an agent's actions. $\alpha \to 0$ corresponds to an agent that will choose an action with the highest expected overall reward. This limit will also reproduce the Bellman-equations since $\log \int \exp(\frac{1}{\alpha} Q(a_t,s_t))da_t \to \max_{a_t} Q(a_t,s_t)$ for $\alpha \to 0$. Therefore, in the maximum-entropy framework the value function corresponds to a "soft" maximization of the Q-function instead of an actual maximization.

An interpretation of equation 1 is as follows: An action a_t is sampled proportional to the overall future expected reward that will result after the action is executed and the policy $\pi(a_t|s_t)$ is followed thereafter. In particular, given deterministic environment transitions we may express the distribution over trajectories as

$$p(\tau) \sim \exp\left(\frac{1}{\alpha} \sum_{(a_i,s_i)\in\tau} Q(a_i,s_i)\right) \tag{4}$$

Therefore, sub-optimal trajectories $\tau_i = \{(a_{i1},s_{i1}),...,(a_{iN},s_{iN})\}$ that choose actions with low Q-values are less likely than trajectories that choose high Q-value actions. Trajectories that have similar overall Q values have the same probability irrespective of what they look like. Therefore, the maximum-entropy framework may describe a situation where a pedestrian yields to a car or continues to cross since it is able to incorporate multi-modal behaviour naturally. Solutions in the limit $\alpha \to 0$ fail to account for multi-modality and collapse to a single mode solution that maximizes the overall reward while ignoring solutions with similar expected overall reward. Furthermore, given similar overall rewards the agent maximizes the entropy of its policy by increasing the probability of all trajectories/ actions as long as this does not impact the expected future reward, e.g. when walking along a cliff on a windy day it may not matter if we walk 2m, 5m or 20m away from the edge but it certainly makes a difference if we try to walk on the edge directly (wind increases probability of falling down). We aim to derive policies that maximize entropy where they can (random behaviour) and minimize it where they need to (deterministic behaviour).

A major advantage of the maximum-entropy framework may be the compositionality of the Q-functions [5]. Imagine two tasks that are each solved with a corresponding Q-function Q_a and Q_b resulting in two policies π_a and π_b. How do we derive a policy that solves both tasks at the same time? We simply add up the Q-functions to obtain $Q = Q_1 + Q_2$ leading to a unified policy $\pi \sim \exp(Q_1 + Q_2)$ [5]. Unfortunately, this is an approximation and may fail to obtain a policy that is close to an optimal policy π^* solving both tasks. If we consider two agents that do not interact with each other in any way then we may derive an optimal policy that solves the task of each agent by adding up the Q-functions $Q(a_1,a_2,s_1,s_2) = Q_a(a_1,s_1) + Q_b(a_2,s_2)$. The state-action pairs (a_i,s_i) do not affect each other. An astronaut navigating on the moon and another astronaut navigating on mars at the same time may be described in this way. Both problems can be solved with a combined Q-function since there is no interaction. Though, if both astronauts navigate inside the same space ship the full Q-function includes an interaction. Astronaut a may want to open a hatch to get outside filling the ship with a vacuum. If Astronaut b does not wear a spacesuit already this action will affect his/ her future actions. Therefore, the Q-function may look like $Q(a_1,a_2,s_1,s_2) = Q_a(a_1,s_1) + Q_b(a_2,s_2) + Q_{OpenHatch}(a_1,a_2,s_1,s_2)$. We will describe the interaction of cars and pedestrians in this way where we have an interaction free Q-function that describes interaction free navigation towards a goal (as solved by [2]) and an interaction only Q-function that switches on when the two agents start to interfere with each other (pedestrian yields to car).

B. Compositionality in Soft-Q-Learning

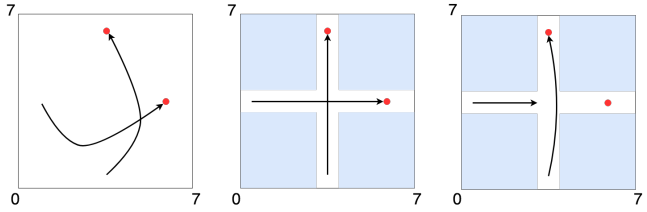


Fig. 2: (**left**) Sub-task 1, the agents try to avoid getting close while approaching their respective goals. (**middle**) Sub-task 2, the agents have to stay outside the blue areas while approaching their respective goals. No collision reward is defined. (**right**) Combined task: the agents stay out of the blue regions, avoid getting close while approaching their goal positions.

We demonstrate that the decomposition of a task into sub-tasks may indeed be feasible for two interacting agents such as a car and a pedestrian. We adapt the didactic example for a multi-goal environment of [7] for our case, using the code provided by the authors. Instead of one agent with multiple goals we have two interacting agents with one goal position each. Additionally, we add obstacles that the agents need to avoid. The rewards that define the behaviour of the agents are as follows: A collision reward of -400 is triggered if the x or y position of the agents gets closer than two distance units; a goal reward of 100 for reaching the goal, which also ends the episode. A reward for going in the correct direction towards the goal is calculated with $400 \cdot \frac{(\vec{s}_t - \vec{goal})\cdot\vec{a}_t}{\|\vec{s}_t - \vec{goal}\|}$. A reward for leaving a specific "lane" of -100, i.e. agent 1 has to stay inside the interval $y \in [-0.5, 0.5]$ and agent 2 needs to stay

inside the interval $x \in [-0.5, 0.5]$.

Mastering the task requires both agents to stay inside their lanes and reach the goal position while avoiding a collision. We will apply soft-Q-learning (SQL) for learning a centralized policy $\pi(a|s)$ that controls both agents at the same time represented by a neural network[1] that samples a four-dimensional continuous action (where to go next) from a Q-function, i.e. $\pi(a|s) \sim \exp(Q(a,s))$ (for details please refer to [7]). It is possible to solve this task by reusing Q-functions from sub-tasks that are inherently different. We will define these sub-tasks as follows (figure 2):

Sub-task 1: *Rewards for approaching the goal position; Negative reward for collision event.*

Sub-task 2: *Rewards for approaching the goal position; Negative reward for moving outside the walkway.*

Sub-task 1 does not involve any lanes/ obstacles, so the agents can roam freely. Contrary to this scenario sub-task 2 does include a reward for leaving the lanes but does not punish collisions. Hence, the agents can get as close to each other as they like. Following the argument in [5] we expect a policy approximating $\pi(a|s) \sim \exp(Q_1(a,s) + Q_2(a,s))$ to perform much better than a policy that is trained on a sub-task $\pi_i(a|s) \sim \exp(Q_i(a,s))$ and tested on the full task (including lane rewards and collision rewards). To verify this hypothesis, we train six (different seeds) Q-functions on sub-task 1 (Q_1), six Q-functions on sub-task 2 (Q_2) and six Q-functions on the combined task for reference (Q). The overall reward is averaged over 10 episodes for each (i.e. 60 evaluations in total). The following table gives an overview of the results (mean reward $\pm$ standard deviation).

Q	$Q_1 + Q_2$	Q_1	Q_2
$\mathbf{1622} \pm 117$	$\mathbf{1657} \pm 62$	-525 ± 48	-546 ± 785

As expected, the combined Q-function outperforms the individual Q-functions significantly and is able to close in on a Q-function that was trained on the full task. Therefore, the combined task can be solved by Q-functions trained on sub-tasks without any retraining on the combined task. The high standard deviation of Q_2 is explained by trajectories that happen to avoid collisions, thus receiving high rewards. It is worth mentioning that adding Q_2 on top of Q_1 improved the overall reward by at least 1247.

Even though the task at hand seems trivial, it takes a certain amount of engineering to successfully derive a policy from the given rewards while avoiding mode collapse, diverging losses and low sample efficiency with the SQL algorithm. This property is also known as "the deadly triad" [21] of function approximation, bootstrapping and off-policy training that is a general observation in reinforcement learning for similar algorithms (e.g. Deep-Q-learning [22]. As such, we will not make any further use of SQL, pointing out a simple alternative for our domain-specific use-case in the next section.

[1]Policy and Q-function are represented using fully connected neural networks with two hidden layers, hidden dimension of 128 and ReLU non-linearities

C. Learning the Interaction Q-Function by Gradient Descent

In reality we are not provided with the rewards that determine how pedestrians interact with other road users. These may be rather complex compared to the simplistic hand engineered rewards used in the previous section which would introduce a bias in terms of what the model can possibly express. We propose to directly propagate the learning signal (i.e. gradient) into the interaction Q-function. The reasoning for why this can be a sensible thing to do is as follows: the beauty of rewards is that they are invariant with respect to new environment settings. E.g. if a goal reward for reaching a goal is given, then the reward does not change, irrespective of where the goal is and if there are any obstacles. The solution and thus the Q-function does change of course. Yet, if a Q-function were only to describe the collision avoidance of two agents given only their current position and velocity, then this Q-function will approximately look the same, irrespective of the environment configuration. The relative distance and velocity are the only important factors (approximately). Therefore, we can simply learn the Q-function Q_2 directly instead of the interaction reward, combine it with a Q-function Q_1 that describes the interaction free behaviour of an agent (i.e. planning) and end up with a Q-function $Q = Q_1 + Q_2$ that solves the overall task. In particular, we assume that the interaction-free Q-function Q_1 is given.

Another advantage of our proposal is that we can circumvent the combination of inverse reinforcement learning and reinforcement learning - both non-trivial steps - to learn the correct rewards and corresponding policy from data. One disadvantage is the potential for compounding errors as long as we only consider one-step actions during training, in contrast to full policy roll-outs in inverse reinforcement learning. Though, this can be compensated for by taking multiple steps into account.

During training we minimize the negative log-likelihood of the observed trajectories via gradient descent (deterministic environment).

$$-\nabla_\psi \log p(\tau|\psi) = -\nabla_\psi \sum_t \log \pi_\psi(a_t|s_t) = \\ -\sum_t \Big[\nabla_\psi Q_\psi(a_t,s_t) - \sum_{\hat{a}_t \in A} \pi_\psi(\hat{a}_t|s_t) \nabla_\psi Q_\psi(\hat{a}_t,s_t)\Big] \quad (5)$$

If our Q-function separates into $Q_\psi(a_t,s_t) = \sum_i Q_{i,\psi_i}(a_t,s_t)$ where every Q_{i,ψ_i} has its own parameters ψ_i, then the gradient w.r.t ψ_i is as follows.

$$-\nabla_{\psi_i} \log p(\tau|\psi) = -\sum_t \Big[\nabla_{\psi_i} Q_{i,\psi_i}(a_t,s_t) - \\ \sum_{\hat{a}_t \in A} \pi_\psi(\hat{a}_t|s_t) \nabla_{\psi_i} Q_{i,\psi_i}(\hat{a}_t,s_t)\Big] \quad (6)$$

1) Including soft-Bellman-equation: One of the main drawbacks of using a flexible Q-function approximation such as a neural network is that it can overfit easily. In particular, the neural network is likely to learn to copy the

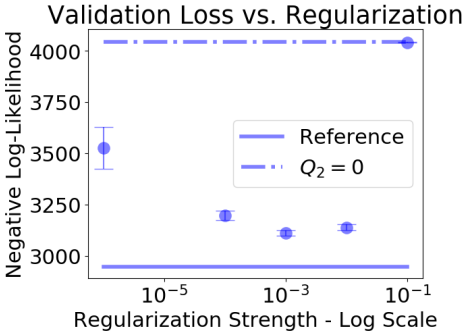


Fig. 3: We approximate Q_2 (interaction) with a neural network and explore different regularization strengths. We provide the mean and standard deviation of 10 training runs. The bottom horizontal line is the (actual) log-likelihood of the validation data that provides a reference for our model. The top horizontal line corresponds to a model where $Q_2 = 0$ so that $Q = Q_1$, thus no interaction is modelled.

exact statistics of the data. Hence, if a certain (s_t, a_t) tuple has only been observed once, then the optimization of the neural network will lead to a deterministic policy that always predicts the same a_t given s_t even though there may be other (s_t, a_t) nearby (i.e. states are very close) that show different behaviour. Additionally, we have no guarantees for the prediction behaviour for states that are outside of the training distribution. There is a simple regularization scheme to address these issues that we can motivate with the soft-Bellman-equation. We assume that the interaction reward r_2 is local, i.e. an agent may only receive a reward if an actual collision happens. This does not mean that the interaction Q-function is local as well. A heavy train needs to start breaking well in advance of the actual collision event, hence the interaction Q-function is "long-range". Though we assume that a pedestrian reacts locally, i.e. the Q-function is "short-range". Therefore, there is a certain distance from the collision event from which on $Q_2 \approx c$ is a constant, thus does not influence the behaviour of the pedestrian. We can see this directly from the soft-Bellman-equation. With ε_{int} being the set of states where $Q_2 \neq 0$.

$$Q(a_t, s_t) = r_1(s_t, a_t) + \overbrace{r_2(s_t, a_t)}^{=0} + \log \int \exp\Big(Q_1(a_t, s_t') + Q_2(a_t, s_t')\Big) da_t \tag{7}$$

$$= r_1(s_t, a_t) + \log \int \exp\Big(Q_1(a_t, s_t') + Q_2(a_t, s_t')\Big) da_t \tag{8}$$

$$\overset{s_t, s_t' \notin \varepsilon_{int}}{\approx} r_1(s_t, a_t) + \log \int \exp\Big(Q_1(a_t, s_t') + c\Big) da_t \tag{9}$$

$$= Q_1(a_t, s_t) + c \tag{10}$$

The last statement corresponds to the soft-Bellman equation for Q_1 shifted by c and is therefore always true. It is noteworthy that shifting the Q-function by a constant c or state-dependent function $c(s_t)$ is a transformation that results in the same policy $\pi(a_t|s_t) \sim \exp(Q(a_t, s_t) + c(s_t)) \sim \exp(Q(a_t, s_t))$ though shifting by $c(s_t)$ is generally not an invariant transformation of the soft-Bellman equation. We can use this to motivate an additional regularization term when training a neural network, enforcing $Q_2 = c = 0$. If the data does not support $Q_2 \neq 0$ (interaction happening), then the neural network should always predict $Q_2 = 0$. We did not see a significant difference in using an l1- or l2-norm and chose the l1-norm for the experiments presented here.

$$g_0 = \lambda \sum_{s \in \Lambda} \sum_{a \in A} |Q_\psi(a, s)| \tag{11}$$

With a hyper-parameter $\lambda \geq 0$ that determines the strength of the regularization. $Q_2 = c$ does not hold true for all states, in particular those close to an actual collision event but will be correct for every other state. If the interactions result in "long-range" Q-functions, this regularization scheme may not be advisable. At least it may be necessary to consider the effective range of the interaction and drop/ weaken the regularization inside that range.

The full optimization target that we want to maximize using Adam [23] is

$$\mathscr{L} = -\sum_{\tau} \sum_{t} \log \pi_{\psi_2}(a_t|s_t) + \lambda \sum_{s \in \Lambda} \sum_{a \in A} |Q_{2,\psi_2}(a, s)| \tag{12}$$

This loss solely depends on the parameters of Q_2 as Q_1 will be known beforehand for the remaining analysis. For the following analysis we choose the interaction-free Q-function Q_1 that [2] proposes to model the path planning of pedestrians in urban environments. The state-space consist of the two-dimensional position of a pedestrian and a two-dimensional action space (e.g. left/right/up/down). Each state receives a reward that depends on the environmental context (e.g. street=-3, walkway=-1) and a goal reward for reaching a pre-defined position. The Q-function is derived by applying equation 2 and 3 over and over again, which is referred to as value iteration.

There are certain limitations to the solution of [2] which we want to point out before progressing further. First, the model does not account for the preferred velocity of the pedestrian, basically defaulting to one average velocity. Second, the model is discrete and thereby introduces a potentially strong bias. Increasing the resolution of the action space and state space scales the time-complexity by $O(|S|^2|A|^2)$. Third, the model has no notion of inertia, due to the fact that the state space is limited to the position of an agent. Without any history of positions (e.g. velocity) the model will not produce smooth trajectories. A major benefit of the maximum-entropy framework is to account for uncertainty. However, the uncertainty should not be modelled by purely positional entropy as [2] proposes but by entropy of higher order derivatives such as the jerk. This would provide us with a variety of smooth trajectories. Using a tabular representation we would need to expand the state space (i.e. include previous states, velocity, ...) which renders the naive application of value iteration intractable.

We want to emphasize that these limitations are not inherent to our proposed algorithm but the choice of Q_1. We use it as it is a simple representation of interaction-free

path planning that helps us to explore our proposal in more detail.

IV. Experiments

A. Unbiased Interaction-Free Q

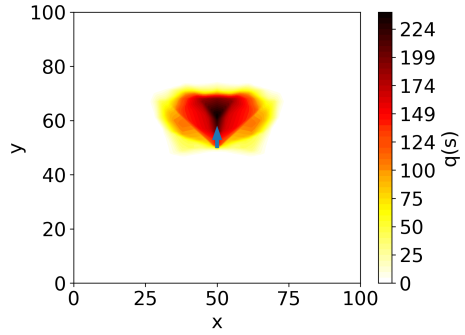
We will analyse the proposed approach on a toy model. This will help us understand the properties of the algorithm. The movement patterns of the agents in our toy model are indicated in figure 5. The car agent can only go upwards on a straight line whereas the pedestrian agent can choose any action within a certain radius around its current position. For simplicity we assume that the states and actions are discrete. The interaction is modelled with a manually constructed Q function:

$$Q_2(a,d) = -\alpha \exp(-\beta \|d\|) \exp(-\gamma |t_1(a,d) - t_2(a,d)|)$$
$$Q_2(a,d) \leq 0 \tag{13}$$

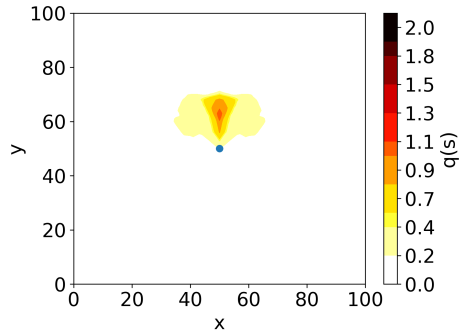
With $\alpha = 1000$, $\beta = 0.2$ and $\gamma = 0.1$. Where d is the relative coordinates $d = s_1 - s_2$. $t_i \in (-\infty, +\infty)$ is the time to collision that we can derive by obtaining the intersection (collision) point of two straight lines given action a and relative coordinates d and calculate the time it takes each agent to get to that point if they were to commit to action a forever. If the lines run in parallel, one of the agents chooses $a_i = 0$ or $t_i < 0$ (past collision point), we set $Q(a,d) = 0$.
The interaction free part is modelled in a similar fashion as [2]. Assuming rewards for getting to the goal position and rewards for staying on the street/ sidewalk results in Q_1 when applying equation 2 and 3 over and over again (value iteration). We set the goal reward to 1, non-goal reward to -15, discount factor to 0.99, grid size to $100x100$ and the maximum action radius to 3.2 for the pedestrian and 3 for the car.
The full Q function is given by adding up the interaction-free and interaction-only part $Q = Q_1 + Q_2$. Given this simple formula our toy model exhibits interesting behaviour such as the car stopping, the pedestrian staying close to one position until the car has passed or the pedestrian swerving behind the car. The policy derived from the Q-function is stochastic so that the behaviour given the same starting position can vary drastically (multimodal behaviour).

Since the Q-function does not depend on any history, we will model it using a simple two-layer fully connected neural network with ReLU activations that takes the relative coordinates d as input and outputs $Q_2(a,d)$.

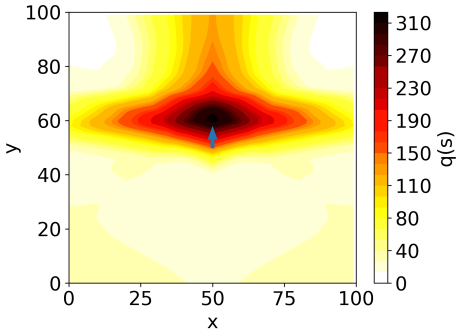
We sampled 100 "expert" trajectories for the training set with the starting x-position of the car being uniformly sampled in the range [40, 60], the starting y-position fixed at 10 and the starting y-position of the pedestrian being sampled in the range [40, 60] and the starting x-position fixed at 80. The validation set consists of 100 trajectories as well, although we did increase the sampling ranges to [20, 90] to create a slight distributional mismatch of the training and validation data set. The goal position of the pedestrian is the state (10, 50), whereas the car is supposed to go all the way



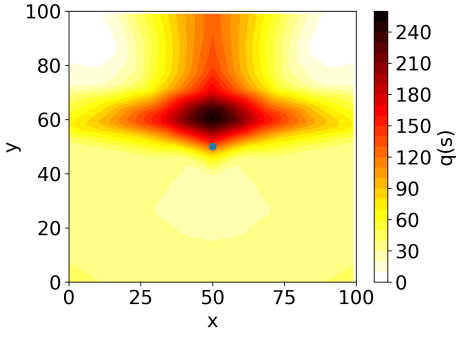
(a) $\lambda = 0.01$, $q(s)$ for all possible pedestrian states, car **moving up** as indicated by blue arrow



(b) $\lambda = 0.01$, $q(s)$ for all possible pedestrian states, car **not moving** as indicated by blue dot. Since no interaction takes place, $q(s)$ is small.

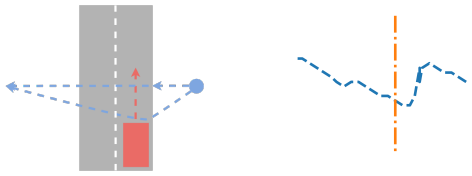


(c) $\lambda = 0$, same as 4a, high degrees of overfitting. Without regularization we fail to account for the local nature of the interaction.



(d) $\lambda = 0$, same as 4b, even though the car is standing still, due to overfitting the unregularized model predicts strong interaction $q(s)$ values for all states.

Fig. 4: Interaction q(s) with and without regularization.



(a) (b)

Fig. 5: (**a**) A pedestrian moves to a goal position on the left side, while the car is moving up. Both try to avoid a collision. (**b**) Example output of actual model with pedestrian swerving behind car.

up (one-dimensional actions only). Since we fully specified a generative model, we actually know the log-likelihood of each trajectory and can use it as a reference for evaluation. Given a learning rate of $\mu = 0.01$ we probe the regularization scheme (11) for multiple regularization strengths. We find that it is important in getting the model predictions close to the reference log-likelihood as can be seen in figure 3. Since the validation loss diverges during training for low regularization strengths ($\lambda < 10^{-4}$), we evaluate 10 training runs (varying seeds) and extract the iteration with the best validation loss. In total we train for 1000 iterations and evaluate on the validation data set on each iteration. With regularization on the other hand, the training converges, with low variance results and the best performance on the validation data set.

The effect of regularization can be seen when investigating the Q_2 values for each state. For this purpose we construct contour plots that indicate whether the neural network predicts anything other then $Q_2 = c(s)$.

$$q(s) = \sum_{a \in A} \left| Q_2(a,s) - \frac{1}{|A|} \sum_{\tilde{a} \in A} Q_2(\tilde{a},s) \right| \tag{14}$$

Basically, we look at the sum of the absolute values of each action. Since $Q_2 = c(s)$ can have high $\sum_{a \in A} |Q_2(a,s)|$ even though it does not affect $\pi(a|s)$ we subtract the mean. If $q(s) \neq 0$ then $\pi(a|s) \sim \exp(Q_1(a,s) + Q_2(a,s)) \nsim \exp(Q_1(a,s))$. Hence, we create a suitable way of visualizing where our model switches on the interaction Q_2 and where it thinks that Q_1 on its own is sufficient. The resulting contour plots for a regularization strength of $\lambda = 0.01$ and no regularization at all are given in fig. 4a-4d. It is important to note that we forced the output of the neural network to be symmetric around the car position as a pedestrian approaching from the left or right is the same.

Since the overall action vector is four-dimensional, we explore two scenarios with fixed car position and action to obtain a meaningful two-dimensional visualization. One scenario where the car is placed at the state (50, 50) with the preselected action $a = [0,0]$ (i.e. do not move) in fig. 4b, 4d and another scenario where it is placed at (50, 50) as well but is going to select the action $a = [0,3]$ (i.e. move up fast) in fig. 4a, 4c. The contour plots indicate where the pedestrian has to pay attention given the car position and car action.

As expected no regularization at all results in a neural network that overfits on the data and predicts significant $q(s)$ values in areas where $Q_2 = 0$ would be the actual solution partially due to not having seen the states in the training distribution. The regularization scheme on the other hand produces sensible contour plots with $q(s) \neq 0$ around the area of where the pedestrian is interacting with the car (state (50, 50)).

B. Biased Interaction-Free Q

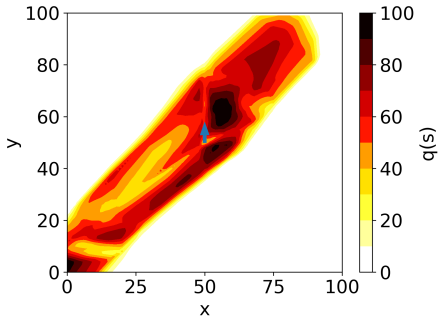


Fig. 6: Due to biased data and biased $Q_1 = 0$ during training, the neural network cannot learn a decent local representation of Q_2.

The reader may ask what happens if we are not provided with the correct interaction-free Q-function Q_1 but with a biased version. Are we still able to obtain the interaction Q-function Q_2? We illustrate the situation by setting $Q_1 = 0$ during training. Now the neural network does not only try to predict the interaction but also the interaction-free movement. Since the data is biased towards a pedestrian moving from right to left and a car moving from the bottom up, the neural network can pick up these patterns even though it is only provided with the relative coordinates d. Fig. 6 shows the contour plot of $q(s)$ for a regularization strength of $\lambda = 0.001$ and no symmetric output (otherwise training gets stuck). The fact that there are large areas where $q(s) = 0$ is mostly due to the fact that the training set of 100 trajectories does not cover the full state space, i.e. the regularization forces the output to $Q_2(a,s) = 0$.

V. DISCUSSION

A. Biased Interaction-Free Q

Assuming that both Q_1 and Q_2 receive the same features/ information, we may reason from the analyses above that if Q_1 is an unbiased model of a non-interacting pedestrian - given proper regularization schemes - we will be able to derive a pure interaction-only Q_2 even from biased data. As we have seen, the story changes if Q_1 is biased. Even though Q_2 is only provided with the relative coordinates d and regularized, it approximates the global interaction-free movement. It can be impossible for the neural network to deduce whether an action with low probability $\pi(a_t|s_t) \sim \exp(Q_1(a_t,s_t))$ is due to interaction effects (avoid collision)

or simply not modelled by the biased Q_1 as there is no external interaction/ no interaction "label" provided. This ambiguity is not unique to training the Q_2-function directly but remains if we were to use e.g. inverse reinforcement learning with a complex reward function (e.g. neural network). Introducing model bias via hand-engineered rewards that specifically target collisions and nothing else can of course enable the model to ignore the short-comings of the biased Q_1. Otherwise, we could set $Q_2 = 0$ outside a certain effective "interaction range" or apply data augmentation such as including situations where we are convinced - using "common sense" - that they are interaction-free (e.g. pedestrians moving away from car) to force $Q_2 = 0$ for those trajectories or increase the diversity of the data.

VI. CONCLUSIONS

Decomposing the Q-functions into an interaction-free and interaction-only component is an attractive way to deal with the generalization challenge of "*environment configurations times interaction configurations*". Experiments using soft-Q-learning suggest that it is feasible to separate the Q-functions in this way. Additionally, we find that as long as the interaction-free Q-function describes human path planning well, we may expect to learn the interaction Q-function via gradient descent. Introducing regularization is crucial for generalization to unseen states and in preventing overfitting. Given the shortcomings of current interaction-free Q-functions available in the literature, we want to improve these in future work to better describe human behaviour and thereby extend our proposal to real-world settings.

REFERENCES

[1] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, "Intent-aware long-term prediction of pedestrian motion," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June. IEEE, may 2016, pp. 2543–2549.

[2] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity Forecasting," in *European Conference on Computer Vision*, 2012, pp. 1–14.

[3] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," in *ECCV*, vol. 8694 LNCS, no. PART 6, 2014, pp. 618–633.

[4] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "DESIRE: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 2165–2174.

[5] T. Haarnoja, V. Pong, A. Zhou, M. Dalal, P. Abbeel, and S. Levine, "Composable Deep Reinforcement Learning for Robotic Manipulation," *arXiv preprint arXiv:1803.06773*, mar 2018.

[6] B. D. Ziebart and J. A. Bagnell, "Modeling Interaction via the Principle of Maximum Causal Entropy," in *In Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 1255–1262.

[7] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement Learning with Deep Energy-Based Policies," in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[8] E. Rehder, F. Wirth, M. Lauer, and C. Stiller, "Pedestrian Prediction by Planning using Deep Neural Networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2018, pp. 1–5.

[9] N. Jaipuria, G. Habibi, and J. P. How, "CASNSC: A context-based approach for accurate pedestrian motion prediction at intersections," in *NIPS*, 2017.

[10] P. Trautman and A. Krause, "Unfreezing the Robot: Navigation in Dense, Interacting Crowds," in *IROS*, 2010.

[11] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, jun 2018.

[12] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 961–971.

[13] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially Compliant Mobile Robot Navigation via Inverse Reinforcement Learning," *The International Journal of Robotics Research*, 2016.

[14] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized Non-communicating Multiagent Collision Avoidance with Deep Reinforcement Learning," in *ICRA*, 2017.

[15] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially Aware Motion Planning with Deep Reinforcement Learning," in *IROS*, mar 2017.

[16] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? A study on pedestrian path prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2014.

[17] R. Quintero, J. Almeida, D. F. Llorca, and M. A. Sotelo, "Pedestrian path prediction using body language traits," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2014.

[18] R. Quintero, I. Parra, D. F. Llorca, and M. A. Sotelo, "Pedestrian Intention and Pose Prediction through Dynamical Models and Behaviour Classification," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2015.

[19] A. Rudenko, L. Palmieri, and K. O. Arras, "Joint Long-Term Prediction of Human Motion Using a Planning-Based Social Force Approach," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2018, pp. 1–7.

[20] S. Russell and A. L. Zimdars, "Q-decomposition for reinforcement learning agents," *ICML*, vol. 20, no. 2, p. 656, 2003.

[21] R. S. Sutton and A. G. Barto, *Reinforcement learning : an introduction*. Cambridge MA: The MIT Press, 2018.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, 2015.

[23] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.