CAR-Net: Clairvoyant Attentive Recurrent Network

Anonymous CVPR submission

Paper ID 283

Abstract

We present an interpretable framework for path prediction that learns scene-specific causations behind agents' behaviors. We exploit two sources of information: the past motion trajectory of the agent of interest and a wide top-down view of the scene. We propose a Clairvoyant Attentive Recurrent Network (CAR-Net) that learns "where to look" in the large image when solving the path prediction task. While previous works on trajectory prediction are constrained to either use semantic information or hand-crafted regions centered around the agent, our method has the capacity to select any region within the image, e.g., a far-away curve when predicting the change of speed of vehicles. To study our goal towards learning observable causality behind agents' behaviors, we have built a new dataset made of top view images of hundreds of scenes (e.g., F1 racing circuits) where the vehicles are governed by known specific regions within the images (e.g., upcoming curves). Our algorithm successfully selects these regions, learns navigation patterns that generalize to unseen maps, outperforms previous works in terms of prediction accuracy on publicly available datasets, and provides human-interpretable static scene-specific dependencies.

1. Introduction

Path prediction consists in predicting the future positions of agents (*e.g.*, humans or vehicles) within an environment. It applies to a wide range of domains from autonomous driving vehicles [35], social robot navigation [33], to abnormal behavior detection in surveillance [25, 24, 30]. Observable cues relevant to path prediction can be grouped into dynamic or static information. The former captures the previous motion behavior of all agents within the scene (past trajectories). The latter represents the static scene surrounding the agents [21, 8, 18]. In this work, we want to learn from the static scene-specific dependencies to solve the prediction task. More broadly, we aim to learn the observable causality behind a chosen path. We formulate the task as follows: given the past trajectory of an agent (x-y coordinates of past few seconds) and a large visual input of the scene (top view of the scene), we want to forecast the motion trajectory of the agent over the next seconds. Our model needs to learn "where to look" within a large visual input to perform the prediction task (see Fig. 1).

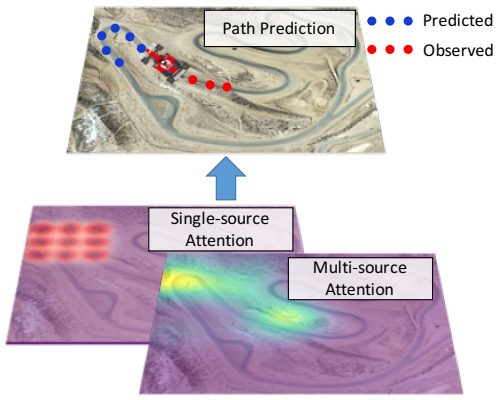

Figure 1: **CAR-Net** is a deep attention-based model that combines two attention mechanisms for path prediction.

Predicting agents' motion while taking into account static scenes is a challenging problem. It requires understanding complex interactions between agents and space, and encoding the corresponding agent-space causalities into the path prediction model. Moreover, scene-specific cues are often sparse and small within the visual input, *e.g.*, a traffic sign within the scene. Finally, these cues might be far away from the agent of interest.

Recent research in computer vision has successfully addressed some of the challenges in path prediction. Kitani et al. [16] have demonstrated that the semantic information about the static environment (*e.g.*, location of sidewalks, extension of grass areas) helps to predict trajectories of pedestrians. However, their method does not rely on raw images to potentially infer finer-grained behaviors. As a result, these methods are limited to scene semantic information rather than just a raw image. Ballan *et al.* [6] have tackled this limitation by modeling human-space interactions using navigation maps which encode how previously observed agents have used each part of the scene. Similarly, Lee *et al.* [19] have used image features to predict agents' paths. However, most of these works have handcrafted their

models to use a small region of fixed size surrounding the agent of interest. They are not able to reason with spatial dependencies that are far away from the target. More generally, none of the existing data-driven methods is able to visualize what the model "uses" within the scene to predict future trajectories, which prevents us from understanding how visual information leads to specific agent behaviors.

We address the limitations of previous path prediction methods by proposing a visual attention model for learning agent-space interactions. Inspired by the recent use of attention models in image captioning [38], machine translation [5] and object recognition [23, 3], we introduce the first visual attention model that can predict the future trajectory of an agent while attending to the salient parts of the scene. Attention based models can be broadly categorized into single and multi-source attention models. The single source attentions (*e.g.*, DRAW [10, 23]) attend to features extracted from a single area of image, while the multi-source attentions (*e.g.*, soft attention from [38]) use a combination of features from multiple areas of an image. In this paper, we propose CAR-Net, a deep neural network architecture which predicts future trajectories - hence being *Clairvoyant* - by processing raw top-view images with a visual *Attentive Recurrent* component. Our attention model combines both single-source and multi-source attention mechanisms. By combining both attention mechanisms, our prediction framework learns a wider spectrum of scene-specific dependencies. Moreover, CAR-Net is simple to implement and train. Hence, it facilitates the use of trajectory prediction in a wide range of other vision tasks such as object tracking [30], activity forecasting [7] and action localization [4].

To study if our proposed architecture is able to learn the observable causality behind agents' behaviors, we built a new dataset where agents' behaviors were driven by known regions within a scene (e.g., a curve in the road). We have collected more than two hundred real world formula one racing circuits and calculated the vehicles' optimal paths given the circuits' curvatures. In this context, the geometry of the road causes the vehicle to speed up or down, and steer. Our attention mechanism succeeds at identifying this causal relationship, and effectively predicts the optimal path of vehicles on these tracks by "looking at the upcoming curve. We further show that the accuracy of our prediction outperforms previous works on the publicly available Stanford Drone Dataset (SDD) where multiple classes of agents (*e.g.*, humans, bicyclists, or buses) navigate outdoor scenes. Our method achieves state-of-the-art results for path prediction given static scene, while providing human-interpretable scene-specific dependencies.

2. Related Work

Trajectory forecasting. Path prediction problem given the dynamic content of a scene have been extensively studied with approaches such as Kalman filters [14], linear regressions [22], or non-linear Gaussian Process [37, 27, 36, 33]. Pioneering work from Helbing and Molnar [26, 17, 12] presented a pedestrian motion model with attractive and repulsive forces referred to as the Social Force model. All these traditional works suffer in modeling complex interactions. Following the recent success of Recurrent Neural Networks (RNN) for sequence prediction tasks, Alahi *et al.* [2] proposed an LSTM model which can learn human movement from the data to predict their future trajectory. Recently, Robicquet *et al.* [28] proposed the concept of social sensitivity with a social force based model to improve path forecasting. Such models suffice for scenarios with few agent-agent interactions and does not consider agent-space interactions. In contrast, we propose methods that can handle more complex environments such as in surveillance of pedestrians, traffic intersections where the locomotion of individual agents is severely influenced by the scene context (*e.g.*, drivable road vs trees and grass).

Recent works have studied how to effectively model the static scene in the prediction task. Kitani *et al.* [16] used the semantic of the scene to forecast plausible paths for a pedestrian using inverse optimal control (IOC). Walker *et al.* in [35] predicted the behavior of generic agents (*e.g.*, a vehicle) in a scene given a large collection of videos. However, they considered a limited number of scenarios. Ballan *et al.* [6] learned scene-specific motion patterns and applied them to novel scenes with an image-based similarity function. However, none of these methods can provide an accurate distant prediction using scene context. Recently, Lee *et al.* [19] proposed a method for the task of long-term future predictions of multiple interacting agents given the scene context. However, all these methods have limited interpretability. our method is instead designed for this specific purpose: communicating why certain behaviors are predicted given the context of the scene.

Visual Attention. Recent works from Xu and Gregor [38, 10] introduce attention based models that learn to attend salient objects related to the task of interest. Xu et al. [38], present soft and hard attention mechanisms that attend to the entire image. Soft attention applies a mask of weights to image's feature maps. Since, the associated training operation is differentiable it has been applied to a wide range of tasks. However, hard attention is not differentiable and it must be trained by Reinforcement Learning. The non-differentiability of this method has led to scarce applications.

Other attention models apply dimensionality reduction to the image. Their goal is to accumulate information over a sequence of partials glimpses of the image. The recurrent attention model introduced in [23] relies on attending a sequence of crops of the image. It has been used in many tasks

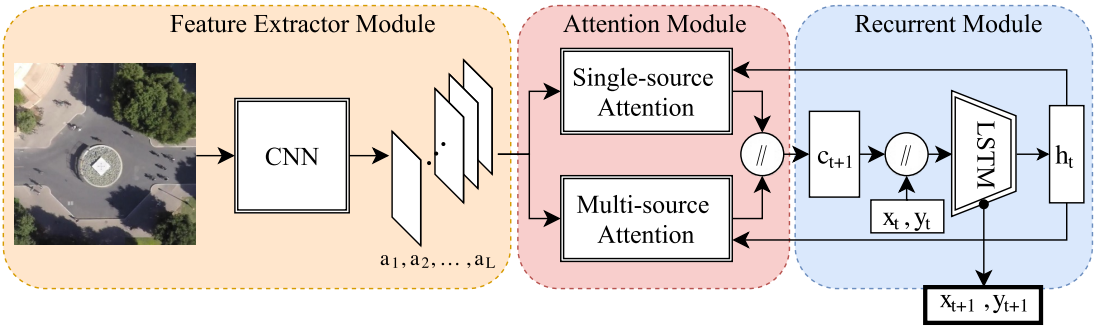

Figure 2: Overview of CAR-Net architecture. Note that "//" block is concatenation operation.

such digits classification and person identification [11, 10]. Visual attention models have also been widely applied to many other applications such as image classification [41], image captioning [38, 40], and video classification [31]. Inspired by these works, we hereby use a visual attention mechanism in our model on trajectory prediction task.

3. CAR-Net

Scene information is necessary to predict the movement of agents. For instance, a cyclist approaching a roundabout changes his path to avoid the collision. Such deviations in trajectories cannot be predicted by only observing the agent's past actions. This motivates us to build a model that can take into account the scene context while predicting an agent's path. In this section, we describe CAR-Net, an attention-based model for path prediction. It performs trajectory prediction using raw top-down images of a scene and the past trajectory of an agent by focusing on the most relevant parts of the images. We first describe the overall architecture. Then, we explain our visual attention module.

3.1. Overall Architecture

The objective of our model is to predict the future path of an agent given its past trajectory and a raw top-down view image of the scene. Our model uses a feature extractor to compute a set of image feature vectors. Then, a visual attention module calculates a context vector c_t representing the salient parts of the image at time t. Finally, in the recurrent module, a long short-term memory (LSTM) network [13] generates the future position of the agent (x_{t+1}, y_{t+1}) at every time step, conditioned on the context vector c_t, the previous hidden state h_t, and the previously generated position of the agent (x_t, y_t). Our model is able to capture agent-space interactions using both the context vector and the past trajectory of the agent.

3.2. Feature extractor module

We extract feature maps from static top-down images using a Convolutional Neural Network (CNN). We use VGGnet-19 [32] pre-trained on ImageNet [29] and fine-tuned on the task of scene segmentation as described in [20]. Fine-tuning VGG on scene segmentation enables the CNN to extract image features that can identify obstacles, roads, sidewalks, and other scene semantics that are essential for trajectory prediction. We use the output of the 5th convolutional layer as image features. The CNN outputs $L = N \times N$ feature vectors $A = \{a_1, ..., a_L\}$ of dimension D, where N and D are size and number of feature maps of the output of the 5th convolutional layer, respectively. Each feature vector corresponds to a certain part of the image. Fig. 2 depicts our feature extractor module.

3.3. Visual attention module

Given a high-dimensional representation of a scene image, we want our model to focus on smaller, discriminative regions of the input image. Using a visual attention method, the most relevant areas of the image are extracted and the irrelevant parts are naturally ignored. The general attention process is as follows. A layer ϕ takes as input the previous hidden state h_t and outputs a vector $\phi(h_t)$ predicting the important areas of image. The vector $\phi(h_t)$ is then applied to the feature map a_t (through a function f_{att}), resulting in a context vector c_{t+1} that contains the important parts of the input image at time step $t+1$:

$$c_{t+1} = f_{att}(A, \phi(h_t)). \quad (1)$$

Our visual attention module can be substituted with any differentiable attention mechanism. Moreover, it can use a combination of several attention methods. Provided that f_{att} and ϕ are differentiable, the whole architecture is trainable by standard back-propagation. We propose three variants for the differentiable attention module that are easily trainable. The first method extracts visual information from multiple areas of the image with a soft-attention mechanism. The second method extracts local visual information from a single cropped area of the image with an attention mechanism inspired by [10]. We refer to the first and second methods as *"multi-source"* and "*single-source*" atten-

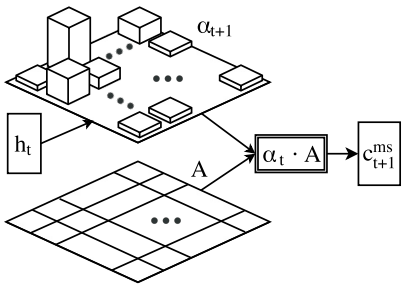

Figure 3: Our multi-source attention mechanism

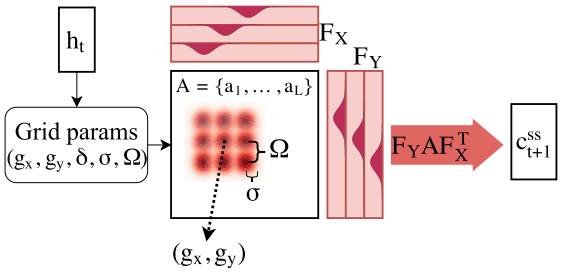

Figure 4: Our single-source attention mechanism

tion mechanisms, respectively. Finally, the attention module of CAR-Net combines both attention mechanisms, allowing our prediction framework to learn a wider spectrum of scene-specific dependencies.

3.3.1 CAR-Net attention

Learning the interactions between agents and space, and encoding the corresponding agent-space causalities into the path prediction model is a challenging task. The scene-specific cues are sometimes sparse and spread in the entire image far away from the agent, or small within a specific area of image. Single and multi-source attention mechanisms attend respectively to global and localized visual information of the scene. When the relevant visual cues are scattered all over the input image, a multi-source attention method can extract a combination of features from multiple key areas of the image. In contrast, when the relevant visual information is localized in one area of the image, single-source attention methods are the good fit to learn to attend to that specific region. In such case, multi-source attention mechanisms are likely to mix essential cues with non-critical image features.

To leverage both local and global visual information in path prediction, the core attention module in CAR-Net combines the two context vectors obtained from single and multi-source attention mechanisms. The combination is done by concatenating the context vectors from single-source c_t^{ss} and multi-source c_t^{ms} attention mechanisms $c_t = [c_t^{ss}, c_t^{ms}]$. The attention module in Fig. 2 depicts the process. More technical details about multi and single-source attention mechanisms can be found in the following Sec. 3.3.2 and 3.3.3, respectively. CAR-Net outperforms both single and multi-source attention mechanisms, proving its ability to leverage the strengths of the two attention mechanisms.

3.3.2 Multi-source attention

The multi-source attention mechanism applies weights to all spatial areas of the scene based on their importance, and outputs a context vector containing relevant scene context from multiple regions of the image. First, the weights matrix α_{t+1} is calculated by passing the hidden state h_t through a fully connected layer ϕ with weight W_{ms}, and bias b_{ms}. Later, the context vector c_{t+1}^{ms} is calculated by element-wise product of the weight matrix α_{t+1} and the feature maps A. Fig. 3 and Eq. 2 show the entire process:

$$\begin{aligned} c_{t+1}^{ms} &= f_{att}(A, \phi(h_t)) \\ &= a \cdot \phi(h_t) = A \cdot \alpha_{t+1} \\ \alpha_{t+1} &= softmax(W_{ms} h_t + b_{ms}). \end{aligned} \tag{2}$$

The soft (multi-source) attention mechanism described in [38] calculates the weight matrix α_t conditioned on both the previous hidden vector and the feature map of the image. However, our α_t relies only on the previous hidden vector. The distinction is important because for path prediction tasks, we do not have the future images of a scene. Consequently, it reduces the computation cost without impacting the performance of the model.

3.3.3 Single-source attention

The single-source attention mechanism illustrated in Fig. 4, attends to a single area of the image. It is inspired by the DRAW attention mechanism, initially designed for the unsupervised setting of digits generation [10]. We adapt it to the supervised learning setting of trajectory prediction. At each time-step $t+1$, five attention parameters $(g_X, g_Y, \delta, \sigma, \gamma)$ are computed via a linear transformation ϕ of the hidden state h_t. They define a local attention patch that is convolved with the feature maps of the image to extract relevant local scene context. More precisely, the attention patch consists of two filter-bank matrices F_X and F_Y, computed entirely with the attention parameters $(g_X, g_Y, \delta, \sigma)$ as in Eq. 4. The filter-bank matrices can be thought of as a grid of N Gaussian filters, of center (g_X, g_Y) and stride δ. The stride of the grid controls the "zoom". As the stride gets larger, it covers a larger area of the original image. Each Gaussian filter in the attention patch has a shared variance σ and mean (ν_x^i, ν_y^i)

$(i \in \{1..N\})$ as expressed in Eq. 3.

$$\begin{aligned} \nu_X^i &= g_X + (i - N/2 - 0.5)\delta \\ \nu_Y^i &= g_Y + (i - N/2 - 0.5)\delta, \end{aligned} \tag{3}$$

$$\begin{aligned} F_X[i,a] &= \frac{1}{Z_X} exp\Big(- \frac{(a - \nu_X^i)^2}{2\sigma^2} \Big) \\ F_Y[j,b] &= \frac{1}{Z_Y} exp\Big(- \frac{(b - \nu_Y^j)^2}{2\sigma^2} \Big). \end{aligned} \tag{4}$$

where (i, j) is a Gaussian in the attention patch, (a, b) is a point in the feature map, and Z_x, Z_y are normalization constants that ensure $\sum_a F_X[i,a] = \sum_b F_Y[j,b] = 1$.

The filter-bank matrices are then convolved over the feature maps A, as in Eq. 5, resulting in a context vector c_{t+1} that is rescaled by a scalar intensity γ. The final context vector contains relevant scene context in the area of the attention patch.

$$\begin{aligned} c_{t+1}^{draw} &= f_{att}(A, \phi(h_t)) \\ &= F_X(h_t)^T A F_Y(h_t). \end{aligned} \tag{5}$$

3.4. Implementation details

We trained the LSTM and the attention modules from scratch with Adam optimizer [15], mini-batch size of 128, and learning rate of 0.001 sequentially decreased every 10 epochs by a factor of 10. All the models are trained for 100 epochs, on the L2 distance between the predicted trajectory and the ground-truth prediction. As in many sequence prediction tasks, the architecture of the model is slightly different at training and test time. At training time, the ground-truth positions are fed as inputs to the recurrent neural network. In contrast, at test time, the predictions of positions (x_t, y_t) are re-injected into the LSTM, as inputs to the next time step.

4. Experiments

We presented CAR-Net, a framework that provides accurate distant future predictions using scene context. We perform a thorough comparison of our method to the state-of-the-art techniques along with comprehensive ablation experiments. We then present insights on the interpretability of our method. We show the generality and robustness of CAR-Net by experimenting with different datasets.

4.1. Data

We tested our models on three datasets that all include trajectory data and top-down images of navigation scenes. We split each dataset into an 80% train, 10% validation, and 10% test sets. Scenes are not shared between sets.

Stanford Drone Dataset (SDD) [28] To show that CAR-Net achieves state-of-the-art performance on path prediction, we tested the model on SDD, a standard benchmark dataset in the path prediction literature [19, 2, 28]. This large-scale dataset consists of top-down videos of various targets (*e.g.*, pedestrians, bicyclists, cars) navigating in the real-world outdoor environment of a university campus (20 different scenes). Agents' trajectories were obtained by the previous tracking. Trajectories were split into segments of 20 time steps (8s total), yielding approximately 230K trajectory segments. Each segment is composed of 8 past positions (3.2s), which are fed to the network as sequential inputs, and 12 future positions (4.8s) used to evaluate the predictions. This is the standard temporal setup for path prediction on SDD. We use raw images to extract visual features, without any semantic labeling. The dataset split we adopt is the standard benchmark split for SDD.

Figure 5: Examples of the scenes captured in our F1 dataset. We annotated each track with the associated optimal racing trajectory. More examples can be found in the supplementary material.

Car-Racing Dataset [1] Studying agent-space dependencies on SDD is complex as agents' behaviors are influenced not only by the semantics of the navigation scene but also by other factors such as interactions between agents. For instance, a pedestrian could decide to stop as they meet an acquaintance. We tested our models on Car-Racing datasets because agents' behaviors can be fully explained by the geometry of the circuit - *e.g.* the curve of an upcoming turn. Our first car racing dataset is composed of 3,000 circuits of various curvatures and road widths that we generated with the Car-Racing-v0 simulator from the OpenAI gym. We simulated two kinds of trajectories on top of these bird-eye view images: (1) trajectories following the median of the road at constant speed and (2) trajectories corresponding to an optimal driving pattern, referred to as "optimal trajectories" and computed with the equations presented in [34]. Note that optimal trajectories have more diverse and complex navigational patterns and include far away scene dependencies compared to constant velocity trajectories. Racing trajectories were split into 24 time-step segments, yielding approximately 500K segments. Similarly to SDD, we used 8 input past positions. For visualization purposes, we evaluated predictions on 16 future positions rather than 12 for SDD.

Formula One Dataset We show that the results obtained on synthetic data generalize well by testing our models on real data. We release a Formula One (F1) dataset composed of real-world car racing circuits and their associated optimal trajectories. In this dataset, the geometry of the road affects agents' behavior (e.g., causes the vehicle to speed up

or down, and steer). The dataset will be available to the public for research purposes and currently includes 250 tracks from different cities of Brazil, Canada, Columbia, Mexico, France, USA and many more countries. Sample circuits are shown in Fig. 5. The top-down images of the tracks were obtained from Google Maps. To generate the racing trajectories, we segmented the roads by hand and computed the associated optimal racing paths. Trajectories were split into 24 time-step segments, 8 input past positions and 16 future positions used for evaluation.

Optimal racing trajectories The ideal racing line is defined as the trajectory around a track that allows a given vehicle to traverse the circuit in the minimum time. Racing lines are represented by periodic smoothed cubic splines. By using a fast, simplified physics simulation, we allow the fitness function to be based on the actual predicted lap time rather than metrics such as total path curvature. The optimal paths provided for our Car-Racing and F1 datasets are based on 2D physics calculated in [9, 34]. The vehicle is represented as a point mass with a total amount of available friction for acceleration in any direction. An iterative process is used to calculate the highest possible speed at each point on a path without the vehicle surpassing its friction.

4.2. Evaluation Metrics and Baselines

We measure the performance of our models on the path prediction task with the following metrics: (i) average displacement error - the mean L2 distance (ML2) over all predicted points of a trajectory and the true points, (ii) final L2 distance error (FL2) - the L2 distance between the predicted final destination and the true final destination at the end of the prediction period T_{pred}.

To perform an ablation study in Section 4.3 and show that our model achieves state-of-the-art performance in Section 4.4, we compare CAR-Net to the following baselines and previous methods from literature.

- *Linear model* (**Lin.**) We use an off-the-shelf linear predictor to extrapolate trajectories with assumption of linear acceleration.
- *Social Force* (**SF**) and *Social-LSTM* (**S-LSTM**). We use the implementation of the Social Force model from [39] where several factors such as group affinity and predicted destinations have been modeled. Since the code for social-LSTM is not available we compare our models with self-implemented version of Social-LSTM from [2].
- *Trajectory Only LSTM* (**T-LSTM**) and *Whole Image LSTM* (**I-LSTM**). These models are simplified versions of our model where we remove the image information and attention module respectively.
- *Multi-Source LSTM* (**MS-LSTM**) and *Single-Source LSTM* (**SS-LSTM**). Our models with only single-source attention or multi-source attention mechanisms respectively.
- *DESIRE*. A deep IOC framework model from [19]. We report the performance of the model *DESIRE-SI-IT0* with top 1 sample.

	Car-Racing Median		Car-Racing Optimal		Formula 1 Optimal	
Model	ML2	FL2	ML2	FL2	ML2	FL2
T-LSTM	10.4	15.5	5.84	10.2	21.2	41.3
I-LSTM	9.71	14.1	5.62	9.5	20.8	40.1
MS-LSTM	7.35	12.7	5.30	8.71	18.9	37.8
SS-LSTM	6.36	9.91	4.64	7.63	14.7	28.9
CAR-Net	**5.0**	**8.87**	**3.58**	**6.79**	**13.3**	**25.8**

Table 1: Performances of our models on the prediction of 16 future positions from 8 past time-steps on the racing circuits datasets: Car-Racing with median and optimal trajectories, and the Formula 1 dataset. We report the ML2 error and the FL2 error - the error at the last time step. CAR-Net tops all models by combining single-source and multi-source attention outputs.

4.3. Ablation Study

We performed an ablation study to show that single-source and multi-source attention mechanisms extract complementary semantic cues from raw images. We analyzed the performances of the baseline models and CAR-Net on the racing circuits datasets (Car-Racing and Formula One datasets). We present the results in table 1.

We observe that the models compare similarly across all racing circuits datasets. First, I-LSTM only slightly outperforms T-LSTM. This is because the circuits' whole feature maps seem to be too complex to significantly complement the dynamic cues extracted from the agents' past trajectories. Second, attention models (MS-LSTM, SS-LSTM, CAR-Net) greatly outperform I-LSTM. This suggests that visual attention mechanisms enhance performance by attending to specific areas of the navigation scenes. We show in Section 4.5 that these attended areas are the relevant semantic elements of the navigation scenes - *e.g.* an upcoming turn. Note that SS-LSTM achieves lower errors than MS-LSTM. This is due to the racing circuits images being large and the relevant semantic cues being mostly located close to the car. Finally, by combining the outputs of single-source and multi-source attention mechanisms, CAR-Net tops MS-LSTM and SS-LSTM on all datasets. We believe this is because the two attention mechanisms attend complementary features.

General remarks. For the Car-Racing dataset, models perform better on the prediction of optimal trajectories than median trajectories. This is because the average pixel distance between consecutive positions is larger for median trajectories, and we trained models on 1K circuits for median trajectories, instead of 3K for optimal trajectories.

4.4. Trajectory Forecasting Benchmark

As table 2 shows, CAR-Net outperforms state-of-the-art methods on the task of predicting 12 future positions (4.8s of motion) from 8 past positions (3.2s) on SDD in both lower ML2 and FL2 error. We report the performance of *DESIRE-SI-IT0 Best* as in [19] on predicting 4s of motion, so the number we report is a lower bound of DESIRE's performance for 4.8s future positions.

Model	ML2	FL2
Lin.	37.11	63.51
SF	36.48	58.14
S-LSTM	31.19	56.97
DESIRE-SI-IT0 Best	29.8	53.25
T-LSTM	31.96	55.27
I-LSTM	30.81	54.21
MS-LSTM	27.38	52.69
SS-LSTM	29.20	63.27
CAR-Net	**25.72**	**51.80**

Table 2: Performance of different baselines on predicting 12 future positions from 8 past time steps on SDD. We report the ML2 error and the FL2 error, in pixels space of the original image. Our method, CAR-Net, achieves by far the lowest error.

T-LSTM baseline achieves a significantly lower ML2 error than the Linear, SF, and S-LSTM models. However, the gap between the FL2 errors of the T-LSTM and SF or S-LSTM models are narrow, suggesting that the T-LSTM model tends to be relatively inaccurate when predicting the last future time-steps. We observe that the SS-LSTM performs poorly compared to the MS-LSTM - especially regarding the FL2 error. We believe multi-source attention performs better due to key semantics in image scenes from SDD being occasionally scattered across the image. In all experiments, CAR-Net outperforms the baselines methods regarding all metrics. Our models outperform the DESIRE single-sample method. This is consistent with [19] mentioning that regression-based models such as CAR-Net are a better fit for use cases where regression accuracy matters more than generating a probabilistic output.

4.5. Qualitative analysis

Visualization details In all figures, the ground-truth and predicted trajectories are plotted in red and blue, respectively. The observed positions are circled in black. We show the multi-source attention weight maps at different points in time, regions where the model attends to are highlighted in white. The attention patches of single-source attention are displayed as bounding boxes. The rectangles indicate the size and location of the attention patches. The center of the patches are represented in yellow.

Short-term predictions Fig. 6 shows the predictions of our models on the datasets studied. On SDD, where key semantic elements are scattered, the multi-source attention mechanism successfully attends multiple relevant areas of the image (top and bottom right image). On racing circuits datasets, we know that the salient semantic elements are the characteristics of the road close to the agent. We observe the multi-source attention model successfully attends to the region around the agent. Yet, this attended region tends to be larger than necessary (top left image).

On the mid-left and mid-center figures, we observe that the first single-source attention patch is off. In the following time steps, the patches jump to a limited area around the agent, identifying the relevant information in the image. The mid-right figure shows an SDD example where the attention patches drift to a region ahead of the car. It seems that a patch on the car would not capture all salient semantics (*e.g.*, the geometry of the upcoming intersection), so the patches reach ahead.

As shown in the bottom row, on Car-Racing and F1 dataset, *CAR-Net* focuses on a narrow region of the image close to the car, using the single-source attention. It is also able to attend to further points like the next curve, using multi-source attention, proving its ability to leverage both attention mechanisms.

Long-term trajectory prediction Fig. 7(a) shows CAR-Net's predictions of 100 consecutive time steps of a median trajectory on the Car-Racing dataset. We observe predictions remain on track. Since first initial observed positions of the car are not important after a while, it proves that the model successfully extracts semantic cues and derives the underlying driving pattern of the ground truth trajectory from them. Note that both single and multi-source attention mechanisms are aligned with predicted positions while attending the salient parts of the scene - *e.g.*, the curve in front of the car.

Agent-space causalities analysis Further experiments illustrate agent-space dependencies. First, we show that road geometry has a large influence on the prediction of future positions. As shown in Fig. 7(b), we manually set the visual attention on a wrong part of the road which was oriented along the top-right direction. We observe that the model predicts positions following a similar top-right axis, while the expected trajectory without any scene information would follow a top-left direction. We can see similar behaviors in the bottom image of Fig. 7(b).

In the second experiment, we show that the model understands which parts of the image are navigable or not. We manually placed the car's past positions outside the racing circuit. Fig. 7(c) shows that the prediction comes back on track and remains stable afterwards. Thus, our model can recover from prediction errors.

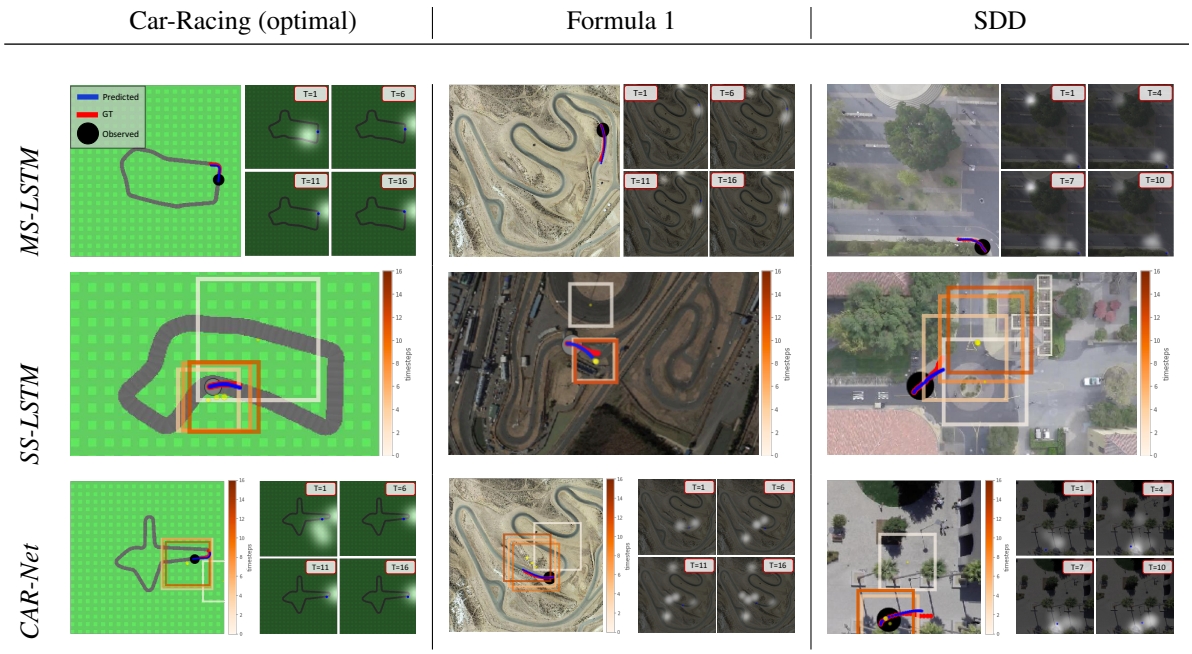

Figure 6: Trained MS-LSTM, SS-LSTM and CAR-Net networks (rows) predicting trajectories on Car-Racing, F1 and SDD datasets (columns). CAR-Net successfully leverages both single-source and multi-source attention mechanisms to predict paths.

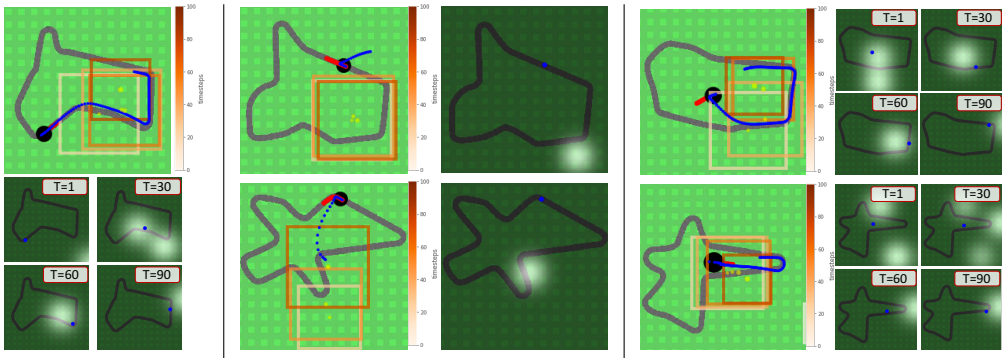

Figure 7: Qualitative analysis: (a) Experiment of CAR-Net on long-term path prediction. Predictions staying on track show that the model learned the driving pattern behind generated trajectories. (b) By manually moving the attention to other parts of the image, we showed that prediction heavily depends on the geometry of the map. (c) When manually imposing the car position to be off-road, the predictions come back on track using the visual cues extracted by the attention mechanism.

5. Conclusions

In this paper, we tackle the trajectory prediction task with CAR-Net, a deep attention-based model that processes past trajectory positions and top-down images of a scene. We propose an attention mechanism that successfully leverages multiple types of visual attention. To study our goal towards learning observable causalities behind agents' behaviors and the scene, we introduce a new dataset made of top view images of hundreds of F1 race tracks where the vehicles' dynamics are governed by specific regions within the images (*e.g.*, the upcoming curve). CAR-Net outperforms baselines on SDD trajectory forecasting benchmark and the new presented F1 car racing dataset by a large margin. By visualizing the output of the attention mechanism, we highlighted agent-space dependencies such as the influence of an upcoming turn on navigation. In future work, we plan to add agent-agent interactions to our framework and to show that our method can be applied to first-view images.

References

[1] OpenAI gym, published = https://gym.openai.com/envs/carracing-v0/, accessed = 2017-01-01. 5

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016. 2, 5, 6

[3] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014. 2

[4] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese. Social scene understanding: End-to-end multi-person action localization and collective activity recognition. *arXiv preprint arXiv:1611.09078*, 2016. 2

[5] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2

[6] L. Ballan, F. Castaldo, A. Alahi, F. Palmieri, and S. Savarese. Knowledge transfer for scene-specific motion prediction. In *European Conference on Computer Vision*, pages 697–713. Springer, 2016. 1, 2

[7] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 2

[8] H. Gong, J. Sim, M. Likhachev, and J. Shi. Multi-hypothesis motion planning for visual object tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 619–626. IEEE, 2011. 1

[9] N. Graham. Smoothing with periodic cubic splines. *Bell Labs Technical Journal*, 62(1):101–110, 1983. 6

[10] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015. 2, 3, 4

[11] A. Haque, A. Alahi, and L. Fei-Fei. Recurrent attention models for depth-based person identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1229–1238, 2016. 3

[12] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 2

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3

[14] R. E. Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. 2

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[16] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012. 1, 2

[17] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2016. 2

[18] H. Kretzschmar, M. Kuderer, and W. Burgard. Learning to predict trajectories of cooperatively navigating agents. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4015–4020. IEEE, 2014. 1

[19] 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. *arXiv preprint arXiv:1704.04394*, 2017. 1, 2, 5, 6, 7

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 3

[21] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(3):397–408, 2005. 1

[22] P. McCullagh. Generalized linear models. *European Journal of Operational Research*, 16(3):285–292, 1984. 2

[23] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014. 2

[24] B. T. Morris and M. M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE transactions on circuits and systems for video technology*, 18(8):1114–1127, 2008. 1

[25] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on*, pages 3153–3160. IEEE, 2011. 1

[26] S. Pellegrini, A. Ess, and L. Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European Conference on Computer Vision*, pages 452–465. Springer, 2010. 2

[27] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005. 2

[28] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016. 2, 5

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 3

[30] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909*, 2017. 1, 2

[31] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015. 3

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[33] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 797–803. IEEE, 2010. 1, 2

[34] R. Vesel. Racing line optimization@ race optimal. *ACM SIGEVOlution*, 7(2-3):12–20, 2015. 5, 6

[35] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3302–3309, 2014. 1, 2

[36] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2008. 2

[37] C. K. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. *Nato asi series d behavioural and social sciences*, 89:599–621, 1998. 2

[38] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. 2, 3, 4

[39] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1345–1352. IEEE, 2011. 6

[40] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4651–4659, 2016. 3

[41] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan. Diversified visual attention networks for fine-grained object classification. *IEEE Transactions on Multimedia*, 19(6):1245–1256, 2017. 3