End-to-end Recurrent Multi-Object Tracking and Trajectory Prediction with Relational Reasoning

Fabian B. Fuchs, Adam R. Kosiorek, Li Sun, Oiwi Parker Jones, Ingmar Posner
Applied AI Lab, University of Oxford
{fabian, adamk, kevin, oiwi, ingmar}@robots.ox.ac.uk

Abstract: The majority of contemporary object-tracking approaches used in autonomous vehicles do not model interactions between objects. This contrasts with the fact that objects’ paths are not independent: a cyclist might abruptly deviate from a previously planned trajectory in order to avoid colliding with a car. Building upon HART, a neural, class-agnostic single-object tracker, we introduce a multi-object tracking method (MOHART) capable of relational reasoning. Importantly, the entire system, including the understanding of interactions and relations between objects, is class-agnostic and learned simultaneously in an end-to-end fashion. We find that the addition of relational-reasoning capabilities to HART leads to consistent performance gains in tracking as well as future trajectory prediction on several real-world datasets (MOTChallenge, UA-DETRAC, and Stanford Drone dataset), particularly in the presence of ego-motion, occlusions, crowded scenes, and faulty sensor inputs. Finally, based on controlled simulations, we propose that a comparison of MOHART and HART may be used as a novel way to measure the degree to which the objects in a video depend upon each other as they move together through time.

Keywords: Multi-Object Tracking, Trajectory Prediction, Relational Reasoning

1 Introduction

Autonomous vehicles need to operate in rich environments that contain a large variety of interacting object. This variety motivates the need for class-agnostic object trackers, which break with the popular tracking-by-detection paradigm [1, 2, 3, 4]. In tracking-by-detection, static video frames are first analysed by an object detector, e.g., a pre-trained deep convolutional neural network (CNN) such as YOLO (Redmon et al. [5]), and then the detected objects are linked across frames. Algorithms from this family can achieve high accuracy, provided sufficient labelled data to train the object detector, and given that all encountered objects can be associated with known classes.

HART is a recently proposed alternative for single-object tracking (SOT), where an arbitrary object can be tracked from an initial video frame (Kosiorek et al. [6]). Since the initial bounding-box is user-provided and may be placed over any part of the image, regardless of whether it corresponds to an object and its class, HART can track arbitrary objects. HART efficiently processes just the relevant part of an image using spatial attention; it also integrates object detection, feature extraction, and motion modelling into one network, which is trained fully end-to-end. Contrary to tracking-by-detection, where only one video frame is typically processed at any given time to generate bounding

Figure 1: Single-object tracking with hierarchical attentive recurrent tracking (HART) (left) and our extension to multi-object tracking (multi-object hierarchical attentive recurrent tracking (MOHART), right). In our proposed framework, the different HART trackers are connected via a relational reasoning module allowing for more robust tracking and more accurate future trajectory prediction.
box proposals, end-to-end learning in HART allows discovering complex visual and spatio-temporal patterns in videos, which is conducive to inferring what an object is and how it moves.

In the original formulation, HART is limited to the single-object modality—as are other existing end-to-end trackers [7, 8, 9]. In this work, we present MOHART, a class-agnostic tracker with complex relational reasoning capabilities provided by a multi-headed self-attention module (Vaswani et al. [10], Lee et al. [11]). MOHART infers the latent state of every tracked object in parallel, and uses self-attention to inform per-object states about other tracked objects. This helps to avoid performance loss under self-occlusions of tracked objects or strong ego-motion. Moreover, since the model is trained end-to-end, it is able to learn how to manage faulty or missing sensor inputs. It can also use the inferred objects’ states to predict their future trajectories, which depend on interactions between different objects. See Figure 1 for a high-level illustration of HART and MOHART.

After describing related work in Section 2 and the methodology in Section 3, we employ the algorithm on toy domains to validate its efficacy in Section 4. By controlling the stochasticity of toy environments, we show that single-object tracking is sufficient in some cases, even those featuring strong long-range interactions, while it may fail in other cases. This may hint at a similar phenomenon in the real world: tracking objects or predicting their future motion independently may be possible in most (but not all) cases, while solving the remaining corner cases might require taking interactions between objects into account. It is these corner cases that motivate our work. In Section 5, we test MOHART on three real world datasets (MOTChallenge [12], UA-DETRAC [13], Stanford Drone dataset [14]) and show that relational reasoning between objects is most important on the MOTChallenge dataset. We hypothesise that this is due to its richness in ego-motion, occlusions and crowded scenes—a result supported by our ablation study. Furthermore, we show that MOHART is able to gracefully handle missing sensory inputs—without any architectural changes. In this case, it falls back on its internal motion model, which also allows for accurate prediction of object locations multiple time steps into the future, learned in a data-driven manner.

2 Related Work

Tracking-by-Detection Vision-based tracking approaches typically follow a tracking-by-detection paradigm: objects are first detected in each frame independently, and then a tracking algorithm links the detections from different frames to propose a coherent trajectory [1, 2, 3, 4]. Motion models and appearance are often used to improve the association between detected bounding-boxes and multiple trackers in a postprocessing step. Recently, elements of this pipeline have been replaced with learning-based approaches such as deep learning [15, 16, 4, 3] or reinforcement learning [17]. Some approaches are targeted towards robustness across domains, for example by using a category-agnostic object detector and performing classification only in a post-processing step [18, 19].

End-to-End Tracking A newly established and much less explored stream of work approaches tracking in an end-to-end fashion. A key difficulty here is that extracting an image crop (according to bounding-boxes provided by a detector), is non-differentiable and results in high-variance gradient estimators. Kahou et al. [7] propose an end-to-end tracker with soft spatial-attention using a 2D grid of Gaussians instead of a hard bounding-box. HART draws inspiration from this idea, employs an additional attention mechanism, and shows promising performance on the real-world KITTI dataset [6]. HART, which forms the foundation of this work, is explained in detail in Section 3. It has also been extended to incorporate depth information from RGBD cameras [8]. Gordon et al. [9] propose an approach in which the crop corresponds to the scaled up previous bounding-box. This simplifies the approach, but does not allow the model to learn where to look—i.e., no gradient is backpropagated through crop coordinates. To the best of our knowledge, there are no successful implementations of any such end-to-end approaches for multi-object tracking beyond SQAIR (Kosiorek et al. [20]), which works only on datasets with static backgrounds. On real-world data, the only end-to-end approaches correspond to applying multiple single-object trackers in parallel—a method which does not leverage the potential of scene context or inter-object interactions.

Pedestrian trajectory prediction Predicting pedestrian trajectories has a long history in computer vision and robotics. Initial research modelled social forces using hand-crafted features [21, 22, 23, 24] or MDP-based motion transition models [25], while more recent approaches learn from context information, e.g., positions of other pedestrians or landmarks in the environment. Social-LSTM
**Figure 2:** Single-object tracking with HART (top) and our extension to multi-object tracking (MOHART, bottom). We track multiple objects in a scene by applying multiple HART trackers in parallel. The trackers can exchange information about their respective objects using query-key-value attention. To do so, HART is split into HART-a and HART-b (see top row) and we apply the self-attention mechanism in-between the parts.

[26] employs a long short-term memory (LSTM) to predict pedestrian trajectories and uses max-pooling to model global social context. Attention mechanisms have been employed to query the most relevant information, such as neighbouring pedestrians, in a learnable fashion [27, 28, 29]. Apart from relational learning, context [30], periodical time information [31], and constant motion priors [32] have proven effective in predicting long-term trajectories.

Our work stands apart from this prior art by not relying on ground truth tracklets. Instead, it addresses the more challenging task of working directly with visual input, performing tracking, modelling interactions, and, depending on the application scenario, simultaneously predicting future motions. As such, it can also be compared to Visual Interaction Networks (VIN) [33], which use a CNN to encode three consecutive frames into state vectors—one per object—and feeds these into a recurrent neural network (RNN), which has an Interaction Network [34] at its core. More recently, Relational Neural Expectation Maximization (R-NEM) has been proposed as an unsupervised approach which combines scene segmentation and relational reasoning [35]. Both VINS and R-NEM are able to make accurate predictions in physical scenarios, but, to the best of our knowledge, have not been applied to real world data.

### 3 Recurrent Multi-Object Tracking with Self-Attention

We start by describing the hierarchical attentive recurrent tracking (HART) algorithm [6], and then follow with an extension of HART to tracking multiple objects, where multiple instances of HART communicate with each other using multi-headed attention to facilitate relational reasoning. We also explain how this method can be extended to trajectory prediction instead of just tracking.

**Hierarchical Attentive Recurrent Tracking (HART)** HART is an attention-based recurrent algorithm, which can efficiently track single objects in a video. It uses a spatial attention mechanism to extract a glimpse $g_t$, which corresponds to a small crop of the image $x_t$ at time-step $t$, containing the object of interest. This allows it to dispense with the processing of the whole image and can significantly decrease the amount of computation required. HART uses a CNN to convert the glimpse $g_t$ into features $f_t$, which then update the hidden state $h_t$ of a LSTM core. The hidden state is used to estimate the current bounding-box $b_t$, spatial attention parameters for the next time-step $a_{t+1}$, as well as object appearance. Importantly, the recurrent core can learn to predict complicated motion conditioned on the past history of the tracked object, which leads to relatively small attention glimpses—contrary to CNN-based approaches (Held et al. [36], Valmadre et al. [37]), HART does not need to analyse large regions-of-interest to search for tracked objects. In the original paper, HART
processes the glimpse with an additional ventral and dorsal stream on top of the feature extractor. Early experiments have shown that this does not improve performance on the MOTChallenge dataset, presumably due to the oftentimes small objects and overall small amount of training data. Figure 2 illustrates HART, further details are provided in Appendix A.

The algorithm is initialised with a bounding-box$^1$ $b_1$ for the first time-step, and operates on a sequence of raw images $x_{1:T}$. For time-steps $t \geq 2$, it recursively outputs bounding-box estimates for the current time-step and predicted attention parameters for the next time-step. The performance of both algorithms is measured as intersection-over-union (IoU) averaged over all time steps in which an object is present, excluding the first time step.

Although HART can track arbitrary objects, it is limited to tracking one object at a time. While it can be deployed on several objects in parallel, different HART instances have no means of communication. This results in performance loss, as it is more difficult to identify occlusions, ego-motion and object interactions. Below, we propose an extension of HART which remedies these shortcomings.

Multi-Object Hierarchical Attentive Recurrent Tracking (MOHART) Multi-object support in HART requires the following modifications. Firstly, in order to handle a dynamically changing number of objects, we apply HART to multiple objects in parallel, where all parameters between HART instances are shared. We refer to each HART instance as a tracker. Secondly, we introduce a presence variable $p_t,m$ for object $m$. It is used to mark whether an object should interact with other objects, as well as to mask the loss function (described in [6]) for the given object when it is not present. In this setup, parallel trackers cannot exchange information and are conceptually still single-object trackers, which we use as a baseline, referred to as HART (despite it being an extension of the original algorithm). Finally, to facilitate communication between trackers, we augment HART with an additional step between feature extraction and the LSTM.

Let $f_{t,m}$ be the feature vector extracted from the glimpse corresponding to the $m^{th}$ object, and let $f_{1:M}$ be the set of such features extracted from all glimpses. Since different objects can interact with each other, it is necessary to use a method that can inform each object about the effects of their interactions with other objects. Moreover, since features extracted from different objects comprise a set, this method should be permutation-equivariant, i.e., the results should not depend on the order in which object features are processed. Therefore, we use the multi-head self-attention block (SAB, Lee et al. [11]), which is able to account for higher-order interactions between set elements when computing their representations, thereby allowing rich information exchange, and it can do so in a permutation-equivariant manner. Intuitively, in our case, SAB allows any of the trackers to query other trackers about attributes of their respective objects, e.g., distance between objects, their direction of movement, or their relation to the robot. This is implemented as follows,

$$Q = W_q f_{1:M} + b_q, \quad K = W_k f_{1:M} + b_k, \quad V = W_v f_{1:M} + b_v,$$  \hspace{0.5cm} (1)

$$O_i = \text{softmax}(Q_i K_i^T) V_i, \quad i = 1, \ldots, H,$$  \hspace{0.5cm} (2)

$$o_{1:M} = O = \text{concat}(O_1, \ldots, O_H),$$  \hspace{0.5cm} (3)

where $o_m$ is the output of the relational reasoning module for object $m$. Time-step subscripts are dropped to decrease clutter. In Equation (1), each of the extracted features $f_{t,m}$ is linearly projected into a triplet of key $k_{t,m}$, query $q_{t,m}$ and value $v_{t,m}$ vectors. Together, they comprise $K, Q$ and $V$ matrices with $M$ rows and $d_q, d_k, d_v$ columns, respectively. $K, Q$ and $V$ are then split up into multiple heads $H \in \mathbb{N}_+$, which allows to query different attributes by comparing and aggregating different projection of features. Multiplying $Q_i K_i^T$ in Equation (2) allows to compare every query vector $q_{t,m,i}$ to all key vectors $k_{t,1:M,i}$, where the value of the corresponding dot-products represents the degree of similarity. Similarities are then normalised via a softmax operation and used to aggregate values $V$. Finally, outputs of different attention heads are concatenated in Equation (3). SAB produces $M$ output vectors, one for each input, which are then concatenated with corresponding inputs and fed into separate LSTMs for further processing, as in HART—see Figure 2.

MOHART is trained fully end-to-end, contrary to other tracking approaches [1, 2, 3, 4]. It maintains a hidden state, which can contain information about the object’s motion. One benefit is that in order to predict future trajectories, one can simply feed black frames into the model. Our experiments show that the model learns to fall back on the motion model captured by the LSTM in this case.

\footnote{We can use either a ground-truth bounding-box or one provided by an external detector; the only requirement is that it contains the object of interest.}
**Figure 3:** HART single object tracking applied four times in parallel. Dashed lines indicate spatial attention, solid lines are predicted bounding boxes at time step $T + 3$, faded circles show the ground truth location at $T + 3$. The repulsive force between each object pair scales with distance as $1/r$. There is no information exchange between the trackers and each tracker evidently only ‘attends’ to its own object. The fact that the future location is predicted accurately (i.e., much better than linear extrapolation) indicates that HART is able to capture complex motion patterns essentially allowing to draw conclusions about the force field. Shown are consecutive time steps from left to right.

**Figure 4:** HART (top, 46% IoU) vs. MOHART (bottom, 76% IoU). Dashed lines show spatial attention, solid lines show predicted bounding boxes, faded circles indicate future ground truth locations. Circles of the same colour repel each other, circles of different colours attract each other. The colour coded identities are randomly assigned in each time step rendering information exchange between trackers (i.e. relational reasoning) necessary. The numbers in the bottom row indicate the self-attention weights from the perspective of the top left tracker (yellow number box).

### 4 Validation on Simulated Data

To test the efficacy of the proposed algorithm, we conduct experiments on a toy domain. First, we show that HART as an end-to-end single-object tracker is able to capture complex motion patterns and leverage these to make accurate predictions. Second, we create a scenario which is not solvable for a single object tracker as it requires knowledge about the state of the other objects and relational reasoning. We show that MOHART, using self-attention for relational reasoning, is able to capture these interactions with high accuracy and compare it to other possible implementations of MOHART (e.g., using max-pooling instead of self-attention). In order to accurately investigate the model’s understanding of motion patterns and interactions between objects, in contrast to traditional tracking, the model is not trained to predict the current location of the object, but its location in a future time step. The domain we create for this purpose is a two dimensional squared box. It contains circular objects with approximated elastic collisions (energy and momentum conservation) between objects and with walls (see Figures 3 and 4).

In the first scenario (Figure 3), four circles each exert repulsive forces on each other, where the force scales with $1/r$, $r$ being their distance. HART is applied four times in parallel and is trained to predict the location of each circle three time steps into the future. The different forces from different objects lead to a non-trivial force field at each time step. Predicting the future location just using the previous motion of one object (Figure 3 shows that each spatial attention box covers only the current object) accurately is therefore challenging. Surprisingly, the single object tracker solves this task with an average of 95% IoU over sequences of 15 time steps. This shows the efficacy of end-
to-end tracking to capture complex motion patterns and use them to predict future locations. This, of course, could also be used to generate robust bounding boxes for a tracking task.

The second scenario (Figure 4) is constructed to be impossible to solve without exchanging information between objects. This is achieved by introducing two colour-coded identities. Agents of the same identity repel each other, agents of different identities attract each other. Crucially, each agent is randomly assigned its identity in each time step. Hence, the algorithm can no longer infer the forces exerted on one object without knowledge of the state of the other objects in the current time step. The forces in this scenario scale with $1/\sqrt{r}$ and the algorithm was trained to predict one time step into the future. HART is indeed unable to predict the future location of the objects accurately (Figure 4 - top). The achieved average IoU is 47%, which is only slightly higher than predicting the objects to have the same position in the next time step as in the current one (34%). A possible interpretation of the qualitative results (green boxes in Figure 4 - top) is that the model uses the momentum of each object to extrapolate into the future. This sometimes works well (bottom right object in frame 31) and sometimes not (top right object in frame 30). Using the relational reasoning module (Figure 4 - bottom), the model is now able to make meaningful predictions (76% IoU). Interestingly, in each frame, the attention scores have a strong correlation with the interaction strength (which directly scales with distance). Despite this not being necessary for the relational reasoning module, this is an interesting side-product as it did not receive any direct supervision.

Figure 5 (left) shows a quantitative comparison of augmenting HART with different relational reasoning modules when identities are re-assigned in every timestep (randomness = 1.0). Exchanging information between trackers of different objects in the latent space with an MLP leads to slightly worse performance than the SOT baseline, while simple max-pooling performs significantly better ($\Delta$IoU $\sim$ 17%). This can be explained through the permutation invariance of the problem: the list of latent representation of the different objects has no meaningful order and the output of the model should therefore be invariant to the ordering of the objects. The MLP is in itself not permutation invariant and therefore prone to overfit to the (meaningless) order of the objects in the training data. Max-pooling, however, is permutation invariant and can in theory, despite its simplicity, be used to approximate any permutation invariant function - given a sufficiently large latent space [38, 39]. Max-pooling is often used to exchange information between different tracklets, e.g., in the trajectory prediction domain [26, 40]. However, self-attention, allowing for learned querying and encoding of information, solves the relational reasoning task significantly more accurately. In Figure 5 (right), the frequency with which object identities are reassigned randomly is varied. The results show that, in a deterministic environment, tracking does not necessarily profit from relational reasoning - even in the presence of long-range interactions. The less random, the more static the force field is and a static force field can be inferred from a small number of observations (see Figure 3). This does of course not mean that all stochastic environments profit from relational reasoning. What these experiments indicate is that tracking can not be expected to profit from relational reasoning by default in any environment, but instead in environments which feature (potentially non-deterministic) dynamics and predictable interactions.
5 Relational Reasoning in Real-World Tracking

![Tracking examples of both HART and MOHART. Coloured boxes are bounding boxes predicted by the model, arrows point at challenging aspects of the scenes. (A) & (C): Each person being tracked is temporarily occluded by a woman walking across the scene (blue arrows). MOHART, which includes a relational reasoning module, handles this more robustly (compare red arrows).](image)

Having established that MOHART is capable of performing complex relational reasoning, we now test the algorithm on three real world datasets and analyse the effects of relational reasoning on performance depending on dataset and task. We find consistent improvements of MOHART compared to HART throughout. Relational reasoning yields particularly high gains for scenes with ego-motion, crowded scenes, and simulated faulty sensor inputs.

5.1 Experimental Details

We investigate three qualitatively different datasets: the MOTChallenge dataset [12], the UA-DETRAC dataset [13], and the Stanford Drone dataset [14]. In order to increase scene dynamics and make the tracking/prediction problems more challenging, we sub-sample some of the high framerate scenes with a stride of two. Training and architecture details are given in Appendix A and Appendix B. We conduct experiments in three different modes:

Tracking. The model is initialised with the ground truth bounding boxes for a set of objects in the first frame. It then consecutively sees the following frames and predicts the bounding boxes. The sequence length is 30 time steps and the performance is measured as intersection over union (IoU) averaged over the entire sequence excluding the first frame. This algorithm is either applied to the entire dataset or subsets of it to study the influence of certain properties of the data.

Camera Blackout. This simulates unsteady or faulty sensor inputs. The setup is the same as in Tracking, but sub-sequences images are blacked out. The algorithm is expected to recognise that no new information is available and that it should resort to its internal motion model.

Prediction. Testing MOHART’s ability to capture motion patterns, only the first two frames are shown to the model followed by three black frames. IoU is measured separately for each time step.

5.2 Results and Analysis

On the MOTChallenge dataset, HART achieves 66.6% intersection over union (see Table 1), which in itself is impressive given the small amount of training data of only 5225 training frames and no pre-training. MOHART achieves 68.5% (both numbers are averaged over 5 runs, independent samples t-test resulted in p < 0.0001). The performance gain increases when only considering ego-motion data. This is readily explained: movements of objects in the image space due to ego-motion scenarios are correlated and can therefore be better understood when combining information from movements of multiple objects, i.e. performing relational reasoning. In another ablation, we filtered for only crowded scenes by requesting five objects to be present for, on average, 90% of the frames in a sub-sequence. For the MOT-Challenge dataset, this only leads to a minor increase of
Table 1: Tracking performance on the MOTChallenge dataset measured in IoU.

|                  | Entire Dataset | Only Ego-Motion | No Ego-Motion | Crowded Scenes | Camera Blackout |
|------------------|----------------|-----------------|---------------|----------------|-----------------|
| MOHART           | 68.5%          | 66.9%           | 64.7%         | 69.1%          | 63.6%           |
| HART             | 66.6%          | 64.0%           | 62.9%         | 66.9%          | 60.6%           |
| Δ                | 1.9%           | 2.9%            | 1.8%          | 2.2%           | 3.0%            |

Table 2: UA-DETRAC Dataset

|                  | All | Crowded Scenes | Camera Blackout |
|------------------|-----|----------------|-----------------|
| MOHART           | 68.1%         | 69.5%           | 64.2%           |
| HART             | 68.4%         | 68.6%           | 53.8%           |
| Δ                | -0.3%         | 0.9%            | 0.4%            |

Table 3: Stanford Drone Dataset

|                  | All | Camera Blackout | CamBlack Bikes |
|------------------|-----|-----------------|----------------|
| MOHART           | 57.3% | 52.6%          | 53.3%          |
| HART             | 56.1% | 53.3%          | 50.7%          |
| Δ                | 1.2%  | 0.7%           | 2.6%           |

the performance gain of MOHART indicating that the dataset exhibits a sufficient density of objects to learn interactions. The biggest benefit from relational reasoning can be observed in the camera blackout experiments (setup explained in Section 5.1). Both HART and MOHART learn to rely on their internal motion models when confronted with black frames and propagate the bounding boxes according to the previous movement of the objects. It is unsurprising that this scenario profits particularly from relational reasoning. Qualitative tracking and camera blackout results are shown in Figure 6 and in Appendix C, respectively.

Tracking performance on the UA-DETRAC dataset only profits from relational reasoning when filtering for crowded scenes (see Table 2). The fact that the performance of MOHART is slightly worse on the vanilla dataset (Δ = −0.3%) can be explained with more overfitting. As there is no exchange between trackers for each object, each object constitutes an independent training sample.

The Stanford drone dataset (see Table 3) is qualitatively different to the other two as it is filmed from a top down view. The scenes are more crowded and each object only covers a small number of pixels rendering it a difficult problem for tracking. The dataset was designed for trajectory prediction, a problem setup where an algorithm is typically provided with ground truth tracklets in coordinate space and potentially an image as context information. The task is then to extrapolate these tracklets into the future. The tracking performance profits from relational reasoning more than on the UA-DETRAC dataset but less than on the MOTChallenge dataset. The performance gain on the camera blackout experiments are particularly strong when only considering cyclists.

In the results from the prediction experiments (see Figure 7) MOHART consistently outperforms HART. On both datasets, the model outperforms a baseline which uses momentum to linearly extrapolate the bounding boxes from the first two frames. This shows that even from just two frames, the model learns to capture motion models which are more complex than what could be observed from just the bounding boxes (i.e. momentum), suggesting that it uses visual information (HART & MOHART) as well as relational reasoning (MOHART). The strong performance gain of MOHART compared to HART on the UA-DETRAC dataset, despite the small differences for tracking on this dataset, can be explained as follows: this dataset features little interactions but strong correlations in motion. Hence when only having access to the first two frames, MOHART profits from estimating the velocities of multiple cars simultaneously.

6 Conclusion

With MOHART, we introduce an end-to-end multi-object tracker that is capable of capturing complex interactions and leveraging these for precise predictions as experiments both on toy and real world data show. However, the experiments also show that the benefit of relational reasoning strongly depends on the nature of the data. The toy experiments showed that in an entirely deterministic world relational reasoning was much less important than in a stochastic environment. Amongst
(A) Prediction results on the MOTChallenge dataset [12].

|       | 0   | 1   | 2   | 3   |
|-------|-----|-----|-----|-----|
| MOHART| 86.4% | 80.0% | 74.0% | 68.4% |
| HART  | 85.3% | 79.2% | 73.1% | 67.4% |
| Momentum | - | - | 78.2% | 70.9% |

(b) Prediction results on the UA-DETRAC dataset (crowded scenes only) [13].

FIGURE 7: Peeking into the future. Only the first two frames are shown to the tracking algorithm followed by three black frames. MOHART learns to fall back on its internal motion model when no observation (i.e. only a black frame) is available. The reported IoU scores show the performance for the respective frames 0, 1, 2, and 3 time steps into the future.

the real-world dataset, the highest performance gains from relational reasoning were achieved on the MOTChallenge dataset, which features crowded scenes, ego-motion and occlusions.

Acknowledgements

We thank Stefan Saftescu for his contributions, particularly for integrating the Stanford Drone Dataset, and Adam Golinski as well as Stefan Saftescu for proof-reading. This research was funded by the EPSRC AIMS Centre for Doctoral Training at Oxford University and an EPSRC Programme Grant (EP/M019918/1). We acknowledge use of Hartree Centre resources in this work. The STFC Hartree Centre is a research collaborative in association with IBM providing High Performance Computing platforms funded by the UK’s investment in e-Infrastructure. The Centre aims to develop and demonstrate next generation software, optimised to take advantage of the move towards exa-scale computing.
References

[1] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. CVPR, 2008.

[2] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. PAMI, 2014.

[3] S.-H. Bae and K.-J. Yoon. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[4] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.

[5] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. Conference on Computer Vision and Pattern Recognition, 2016.

[6] A. R. Kosiorek, A. Bewley, and I. Posner. Hierarchical attentive recurrent tracking. Neural Information Processing Systems, 2017.

[7] S. E. Kahou, V. Michalski, and R. Memisevic. RATM: recurrent attentive tracking model. IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017.

[8] M. Rasouli Danesh, S. Yadav, S. Herath, Y. Vaghei, and S. Payandeh. Deep attention models for human tracking using rgbd. Sensors, 19:750, 02 2019.

[9] D. Gordon, A. Farhadi, and D. Fox. Re3 : Real-Time Recurrent Regression Networks for Visual Tracking of Generic Objects. RA-L, 2018.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. Neural Information Processing Systems, 2017.

[11] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh. Set transformer. In International Conference on Machine Learning, 2019.

[12] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. 2016. arXiv: 1603.00831.

[13] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu. DETRAC: A new benchmark and protocol for multi-object tracking. arXiv, 1511.04136, 2015.

[14] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory prediction in crowded scenes. European Conference on Computer Vision, 2016.

[15] H. Nam and B. Han. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. CVPR, 2016.

[16] G. Ning, Z. Zhang, C. Huang, Z. He, X. Ren, and H. Wang. Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking. ISCAS, 2017.

[17] Y. Xiang, A. Alahi, and S. Savarese. Learning to Track: Online Multi-Object Tracking by Decision Making Multi-Object Tracking. ICCV, 2015.

[18] A. Ošep, W. Mehner, P. Voigtlaender, and B. Leibe. Track, then Decide: Category-Agnostic Vision-based Multi-Object Tracking. ICRA, 2018.

[19] P. Ondruska and I. Posner. Deep Tracking: Seeing Beyond Seeing Using Recurrent Neural Networks. AAAI, 2016.

[20] A. Kosiorek, H. Kim, Y. W. Teh, and I. Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In Advances in Neural Information Processing Systems, pages 8606–8616, 2018.
[21] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In Computer Graphics Forum, 2007.

[22] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You’ll never walk alone: Modeling social behavior for multi-target tracking. In ICCV 2009.

[23] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In IROS, 2010.

[24] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In CVPR, 2011.

[25] 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, 2018.

[26] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In CVPR, 2016.

[27] H. Su, Y. Dong, J. Zhu, H. Ling, and B. Zhang. Crowd scene understanding with coherent recurrent neural networks. 2016.

[28] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. Neural networks, 2018.

[29] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019.

[30] D. Varshneya and G. Srinivasaraghavan. Human trajectory prediction using spatially aware deep attention models. arXiv preprint:1705.09436, 2017.

[31] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett. 3dof pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data. In 2018 IEEE International Conference on Robotics and Automation. IEEE, 2018.

[32] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll. The simpler the better: Constant velocity for pedestrian motion prediction. arXiv preprint arXiv:1903.07933, 2019.

[33] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti. Visual Interaction Networks: Learning a Physics Simulator from Video. NIPS, 2017.

[34] P. W. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu. Interaction Networks for Learning about Objects, Relations and Physics. NIPS, 2016.

[35] S. van Steenkiste, M. Chang, K. Greff, and J. Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. ICLR, 2018.

[36] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In European Conference on Computer Vision, 2016.

[37] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. End-to-end representation learning for correlation filter based tracking. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[38] M. Zaheer, S. Kottur, S. Ravanbhakhsh, B. Póczos, R. Salakhutdinov, and A. Smola. Deep Sets. In Advances in Neural Information Processing Systems, 2017.

[39] E. Wagstaff, F. B. Fuchs, M. Engelcke, I. Posner, and M. A. Osborne. On the limitations of representing functions on sets. International Conference on Machine Learning, 2019.

[40] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN: socially acceptable trajectories with generative adversarial networks. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
Figure 8: Camera blackout experiment on a pedestrian street scene from the MOTChallenge dataset without ego-motion. Subsequent frames are displayed going from top left to bottom right. Shown are the inputs to the model (some of them being black frames, i.e., arrays of zeroes) and bounding boxes predicted by MOHART (coloured boxes). This scene is particularly challenging as occlusion and missing sensor input coincide (fourth row).

A Architecture Details

The architecture details were chosen to optimise HART performance on the MOTChallenge dataset. They deviate from the original HART implementation [6] as follows: The presence variable is predicted with a binary cross entropy loss. The maximum number of objects to be tracked simultaneously was set to 5 for the UA-DETRAC and MOTChallenge dataset. For the more crowded Stanford drone dataset, this number was set to 10. The feature extractor is a three layer convolutional network with a kernel size of 5, a stride of 2 in the first and last layer, 32 channels in the first two layers, 64 channels in the last layer, ELU activations, and skip connections. This converts the initial $32 \times 32 \times 3$ glimpse into a $7 \times 7 \times 64$ feature representation. This is followed by a fully connected layer with a 128 dimensional output and an elu activation. The spatial attention parameters are linearly projected onto 128 dimensions and added to this feature representation serving as a positional encoding. The LSTM has a hidden state size of 128. The self-attention unit in MOHART comprises linear projects the inputs to dimensionality 128 for each keys, queries and values. For the real-world experiments, in addition to the extracted features from the glimpse, the hidden states from the previous LSTM state are also fed as an input by concatenating them with the features. In all cases, the output of the attention module is concatenated to the input features of the respective object.

As an optimizer, we used RMSProp with momentum set to 0.9 and learning rate $5 \times 10^{-6}$. For the MOTChallenge dataset and the UA-DETRAC dataset, the models were trained for 100,000 iterations of batch size 10 and the reported IoU is exponentially smoothed over iterations to achieve lower variance. For the Stanford Drone dataset, the batch size was increased to 32, reducing time to convergence and hence model training to 50,000 iterations.

B Experimental Details

The MOTChallenge and the UA-DETRAC dataset discussed in this section are intended to be used as a benchmark suite for multi-object-tracking in a tracking-by-detection paradigm. Therefore, ground truth bounding boxes are only available for the training datasets. The user is encouraged to upload their model which performs tracking in a data association paradigm leveraging the provided bounding box proposals from an external object detector. As we are interested in a different analysis (IoU given initial bounding boxes), we divide the training data further into training and test sequences. To make up for the smaller training data, we extend the MOTChallenge 2017 dataset with three
sequences from the 2015 dataset (ETH-Sunnyday, PETS09-S2L1, ETH-Bahnhof). We use the first 70% of the frames of each of the ten sequences for training and the rest for testing. Sequences with high frame rates (30Hz) are sub-sampled with a stride of two. For the UA-DETRAC dataset, we split the 60 available sequences into 44 training sequences and 16 test sequences. For the considerably larger Stanford Drone dataset we took three videos of the scene deathCircle for training and the remaining two videos from the same scene for testing. The videos of the drone dataset were also sub-sampled with a stride of two to increase scene dynamics.

C Camera Blackout Experiments

In Section 5, we conducted a set of camera blackout experiments to test MOHART’s capability of dealing with faulty sensor inputs. While traditional pipeline methods require careful consideration of different types of corner cases to properly handle erroneous sensor inputs, MOHART is able to capture these automatically, especially when confronted with similar issues in the training scenarios. To simulate this, we replace subsequences of the images with black frames. Figure 8 and Figure 9 show two such examples from the test data together with the model’s prediction. MOHART learns not to update its internal model when confronted with black frames and instead uses the LSTM to propagate the bounding boxes. When proper sensor input is available again, the model uses this to make a rapid adjustment to its predicted location and ‘snap’ back onto the object. This works remarkably well in both the presence of occlusion (Figure 8) and ego-motion (Figure 9). Tables 1 to 3 show that the benefit of relational reasoning is particularly high in these scenarios specifically. These experiments can also be seen as a proof of concept of MOHART’s capabilities of predicting future trajectories—and how this profits from relational reasoning.