Transfer of Status Report
Structured Sequence Modelling

Adam Kosiorek[1]

Supervisor: Prof. Ingmar Posner

I. INTRODUCTION

Sequence prediction is a powerful framework for unsupervised learning. With the abundance of sequential real-world data available, it is possible to train large and highly-complex models for the next time-step prediction. While temporal dependencies facilitate learning in humans, the majority of machine learning (ML) algorithms either do not use them or do so in a very limited sense. Temporal dependencies determine the evolution of the world, and therefore it is possible that models trained for sequence prediction can learn intuitive physics and that they might be useful in model-based reinforcement learning.

While sequence predictors achieve some success, they generally do not provide interpretable representations that could be easily used for downstream tasks, nor do they use any prior information about the problem. Recent advances in variational inference and neural networks allow building efficient generative models of high-dimensional data. They do so by specifying the probability distribution of observed data in terms of (typically low-dimensional) auxiliary latent variables. On one hand, the latent variables explain the observations and can be useful in downstream tasks. On the other hand, the stochasticity of this approach affords better regularisation and uncertainty estimates. The problem of including prior information has not been addressed, however.

All work performed under this thesis will aim at utilising prior information in generative models of time-series and exploiting these models in the context of model-based reinforcement learning. We will argue that prior information can be introduced by structuring the model appropriately, thereby factorising latent variables and enforcing their *specific* semantic meaning. In the process, we will construct a generative model of moving objects, capable of describing the scene with variable-length latent representation. This model will make it possible to learn intuitive physics and simulate future motion and appearance changes of visible objects. Finally, we will use this model in the Dyna

[1]Applied Artificial Intelligence Lab, Oxford Robotics Institute, University of Oxford.

(Sutton, 1991) framework, with the goal of improving sample-efficiency of model-free reinforcement learning algorithms.

The rest of this paper is structured as follows. Section II covers prior work related to the areas in question. I summarise the task of sequence prediction, describe some variants of unsupervised learning and investigate how model structure helps to learn abstract concepts from data. In section III, I describe our work on object tracking and how it ties with my interests and the planned future work on structured unsupervised learning for videos and model-based reinforcement learning. Section IV details my future research plans, related risks and expected outcomes. Section V concludes this work.

II. Related Work

A. Unsupervised Learning via Generative Modelling

While data in general is abundant and cheap, data for supervised learning is often expensive and time-consuming to gather. The majority of ML algorithms require relatively large amounts of labelled training data. One of the explanation states that they start learning without any prior knowledge of the world. This is in stark contrast to humans, who not only have a vast amount of knowledge about the world, but also expand it continuously and without any supervision (Friston, 2009). One alternative is to perform generative modelling of the probability distribution $p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) \,\mathrm{d}\mathbf{z}$ of observations $\mathbf{x}$ in terms of some latent variables $\mathbf{z}$. The latent variables *explain* the observations and can make the joint distribution $p(\mathbf{x}, \mathbf{z})$ tractable even in the case of an intractable marginal distribution. The latent encoding can be used in downstream tasks e. g., for transfer or semi-supervised learning (Pan and Yang, 2010). Hinton, Osindero, and Teh, 2006 introduced Deep Belief Networks (DBN) which explain the observations in terms of Bernoulli latent variables. Alternatively, we can introduce an approximate posterior distribution and approximate the true data distribution by maximising the evidence lower bound (ELBO) on the log probability of the data. This approach results in variational autoencoders (VAE) (Kingma and Welling, 2014; Danilo J Rezende, Mohamed, and Wierstra, 2014). VAEs are much more flexible than DBNs as they allow latent variables from arbitrary probability distribution functions (pdf) and can be trained end-to-end with off-the-shelf gradient-based methods. Performance of VAEs depends on the choice of the approximate posterior distribution and a prior for the latent space. Since the latter is stochastic, the variance of the gradient estimator is increased compared to deterministic neural networks, which leads to slower convergence. These approaches are primarily suited to modelling datasets of independent and identically distributed (*i.i.d.*) points.

B. Sequence Modelling

Traditional approaches to sequence modelling often consider inference of latent variables that explain the data e. g., linear dynamical systems or hidden markov models (Bishop, 2006). They often require dynamics of the system to be known and often have too little capacity to model complex and high-dimensional real-world data. Neural networks, on the other hand, can learn both features and state dynamics from data and they can approximate functions of arbitrary complexity with arbitrary precision. Even early works on the topic demonstrated how useful neural networks are for prediction of chaotic time-series (Lapedes and Farber, 1988). Since then, neural networks have been successfully applied to sequence classification and prediction in different domains: written natural language, speech and audio, motion capture data or brain waves (Längkvist, Karlsson, and Loutfi, 2014). Sequence prediction is a promising method of unsupervised learning. The task is to predict the observation at time $t+1$ given a sequence of observations $\mathbf{x}_{1:t}$ up to time t. It is flexible in that it admits many different model types, including Gaussian processes, support vector machines or feed-forward neural networks, although models which can explicitly use temporal structure of data such as Gaussian process dynamic models (GPDM; Wang, Fleet, and Hertzmann, 2008) or recurrent neural networks (RNN) tend to perform better. Recently, sequential counterparts of VAEs have been proposed, which allow efficient generative modelling of sequences, with the additional advantage of better regularisation and superior uncertainty estimates (Fabius and Amersfoort, 2015; Bayer and Osendorfer, 2015; Karl et al., 2017; Fortunato, Blundell, and Vinyals, 2017). Unlike deterministic RNNs, sequential VAEs model time-series in terms of low-dimensional latent variables that can be used in downstream tasks. Moreover, they allow counter-factual simulation and generation of multiple trajectories from a single starting point due to their stochastic nature. The most relevant prior work is that on state estimation and next-frame prediction in videos. Ondruska and Posner, 2016 introduces Deep Tracking, which aims to estimate state in a partially-observable environment. It uses two-dimensional occupancy grids and predicts grid occupancy in the future. While related, it is unclear how this approach could be used to model any other type of data. Next-frame prediction has been done recently as predictive coding (Lotter, Kreiman, and Cox, 2016; Canziani and Culurciello, 2017). The idea dates back to the Kalman filter (Kalman, 1960) and states that the hidden state of the model should be updated only to remove any discrepancies between the predictions and the observations at the following time-steps. While very general, this approach imposes additional structure on the prediction problem: it (i) removes redundancies from in the input data, (ii) creates an inner feedback loop, which could adjust model dynamics at runtime to minimise any errors and (iii) could implement human-like attention mechanisms if realised probabilistically (Friston, 2009).

C. Learning of Abstract Ideas

The utility of sequence prediction as an unsupervised learning approach can be intuitively explained by the fact that predicting the future, if it is to be done well, requires very good understanding of the present. If, for example, a model can learn an idea of an object and the laws of physics, it should be able to constrain its prediction to those physically plausible: e. g., a car should not dissolve into thin air. The majority of neural models are over-parametrised (Denil et al., 2013), however, which makes learning abstract notions from data extremely sample inefficient. Eslami et al., 2016 introduce Attend, Infer, Repeat (AIR), a VAE with a variable-length latent encoding for image reconstruction. This model imposes a geometric prior on the encoding length which encourages sparse solutions, therefore learning to decompose the scene into a number of independent parts — the objects. It is worth noting that, along the main model, the authors introduce difference-AIR, which exploits the specific structure of the problem and adheres to the predictive coding paradigm, thereby achieving better performance. In the extension of this work, Danilo Jimenez Rezende et al., 2016 learn to reconstruct three-dimensional (3D) structure of an object from even a single two-dimensional (2D) view by imposing 3D latent representation and structuring the decoder as a projection of the latent space into the 2D output space; they show that their model is able to infer the idea of an object from data. Häusser, Mordvintsev, and Cremers, 2017 learn the idea of an object and its class by learning to associate similar objects with each other in the embedding space, which is very much like a child learning about its identity by comparing itself with others (Decety and Chaminade, 2003). In case of reinforcement learning, a complex environment might itself be a cue which leads to learning abstract ideas. Heess et al., 2017 shows that articulated agents can learn real-world motion patterns by interacting with the environment. Specifically, they learn to crouch, jump, turn and run while maximising a very simple reward function based on forward progress. Using a specific model structure as a method of learning abstract ideas was also demonstrated by Battaglia et al., 2016. The authors propose an interaction network, a highly complex model that operates on a graph of objects and relations between them and acts as a physics simulator. The particular model structure enables learning invariants (e. g., energy conservation) and inferring latent variables describing the system as a whole (e. g., potential energy).

In the following we put these ideas together.

III. Submitted Work

During my first year as a DPhil student at Oxford we developed the Hierarchical Attentive Recurrent Tracking (HART) framework. It was submitted to NIPS 2017. This RNN-based model learns to track objects in videos by focusing on small image regions, usually not much bigger than the tracked objects. It does so by using a differentiable attention mechanism, which can effectively crop a part of the image, thereby quickly removing irrelevant parts of the input. Upscaling HART to a challenging real-world dataset proved difficult, as end-to-end training on a randomly initialised neural network was very unstable and converged to poor results. To address this issue, I resorted to transfer learning and used AlexNet (A. Krizhevsky, I. Sutskever, and Hinton, 2012) as a feature extractor, which has stabilised the training and improved performance (*cf.* section 5.2. in the paper). Feature extractors pre-trained on static image analysis tasks are often used for processing video sequences (see e. g., Ning et al., 2016). This approach, while effective, has little justification in neuroscience. In contrary, there is a growing body of evidence indicating the importance of temporal connections in the human visual cortex (Kastner and Ungerleider, 2000), which suggests that the temporal integration of information is vital for building up high resolution representation of the world.

Modern single-object-tracking approaches are based on either metric learning or bounding box regression (Bertinetto et al., 2016 and Held, Thrun, and Savarese, 2016, respectively). Not only do they need to rely on heuristics (non-differentiable image cropping, explicit scale search) to achieve computational efficiency and accuracy, but they are also fully dependent on a single error signal for learning. HART, on the other hand, exploits the model structure to make more efficient use of the error signal. It uses the ground-truth bounding boxes to derive three related but distinct learning signals, one of each for the model parts. One of them, the object masks for foreground-background segmentation of extracted attention glimpses, can be seen as self-supervision. It serves different purposes: (a) it forces the model to store object appearance information in the hidden state, (b) it encourages better spatial attention prediction, as computing the object mask is easier (lower relative penalty for any mistakes) if the object covers a bigger part of the attention glimpse and (c) since the ground-truth object mask is computed on the fly, it serves as data-augmentation, in the sense that the errors and the learning signal is dependent on the model parameters and is different in every iteration of training even if the input data is the same. Despite being trained under full supervision, HART was a test bench I used to learn about learning in the presence of temporal dependencies and to experiment with different structures of the objective function so as to maximise learning from a limited amount of data.

Hierarchical Attentive Recurrent Tracking

Adam R. Kosiorek
Department of Engineering Science
University of Oxford
adamk@robots.ox.ac.uk

Alex Bewley
Department of Engineering Science
University of Oxford
bewley@robots.ox.ac.uk

Ingmar Posner
Department of Engineering Science
University of Oxford
ingmar@robots.ox.ac.uk

Abstract

Class-agnostic object tracking is particularly difficult in cluttered environments as target specific discriminative models cannot be learned *a priori*. Inspired by how the human visual cortex employs spatial attention and separate "where" and "what" processing pathways to actively suppress irrelevant visual features, this work develops a hierarchical attentive recurrent model for single object tracking in videos. The first layer of attention discards the majority of background by selecting a region containing the object of interest, while the subsequent layers tune in on visual features *particular* to the tracked object. This framework is fully differentiable and can be trained in a purely data driven fashion by gradient methods. To improve training convergence, we augment the loss function with terms for a number of auxiliary tasks relevant for tracking. Evaluation of the proposed model is performed on two datasets of increasing difficulty: pedestrian tracking on the KTH activity recognition dataset and the KITTI object tracking dataset.

1 Introduction

In computer vision, the task of class-agnostic object tracking is challenging since no target-specific model can be learnt *a priori* and yet the model has to handle target appearance changes, varying lighting conditions and occlusion. To make it even more difficult, the tracked object often constitutes but a small fraction of the visual field. The remaining parts may contain *distractors*, which are visually salient objects resembling the target but hold no relevant information. Despite this fact, recent models often process the whole image, exposing them to noise and increases in associated computational cost or use heuristic methods to decrease the size of search regions. This in contrast to human visual perception, which does not process the visual field in its entirety, but rather acknowledges it briefly and focuses on processing small fractions thereof, which we dub *visual attention*.

Attention mechanisms have recently been explored in machine learning in a wide variety of contexts [1, 13], often providing new capabilities to machine learning algorithms [10, 11, 7]. While they improve efficiency [21] and performance on state-of-the-art machine learning benchmarks [1], their architecture is much simpler than that of the mechanisms found in the human visual cortex [6]. Attention has also been long studied by neuroscientists [26], who believe that it is crucial for visual perception and cognition [5], since it is inherently tied to the architecture of the visual cortex and can affect the information flow inside it. Whenever more than one visual stimulus is present in the receptive field of a neuron, all the stimuli compete for computational resources due to the limited processing capacity. Visual attention can lead to suppression of distractors, by reducing the size of

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

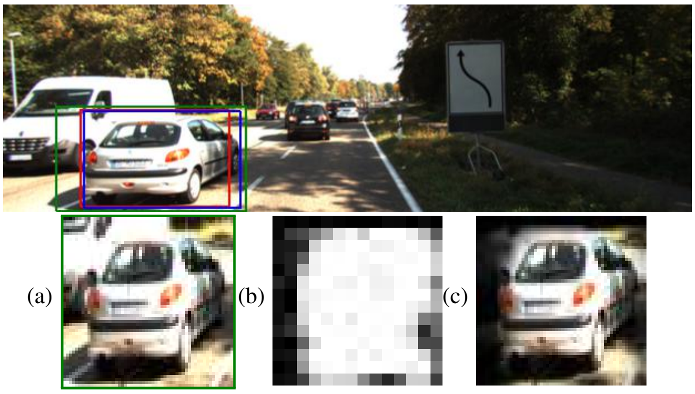


Figure 1: KITTI image with the ground-truth and predicted bounding boxes and an attention glimpse. The lower row corresponds to the hierarchical attention of our model: 1^{st} layer extracts an attention glimpse (a), the 2^{nd} layer uses appearance attention to build a location map (b). The 3^{rd} layer uses the location map to suppress distractors, visualised in (c).

the receptive field of a neuron and by increasing sensitivity at a given location in the visual field (*spatial attention*). It can also amplify activity in different parts of the cortex, which are specialised in processing different types of features, leading to response enhancement w. r. t. those features (*appearance attention*). The functional separation of the visual cortex is most apparent in two distinct processing pathways. After leaving the eye, the sensory inputs enter the prefrontal cortex (known as *V1*) and then split into the *dorsal stream*, responsible for estimating spatial relationships (*where*), and the *ventral stream*, which targets appearance-based features (*what*).

Inspired by the general architecture of the human visual cortex and the role of attention mechanisms, this work presents a biologically-inspired recurrent model for single object tracking in videos (*cf.* section 3). Tracking algorithms typically use simple motion models and heuristics to decrease the size of the search region. It is interesting to see whether neuroscientific insights can aid our computational efforts, thereby improving the efficiency and performance of single object tracking. It is worth noting that visual attention can be induced by the stimulus itself (due to, e. g., high contrast) in a *bottom-up* fashion or by back-projections from other brain regions and working memory as *top-down* influence. The proposed approach exploits this property to create a feedback loop that steers the *three* layers of visual attention mechanisms in our hierarchical attentive recurrent tracking (*HART*) framework, see Figure 1. The first stage immediately discards spatially irrelevant input, while later stages focus on producing deictic filters to emphasise visual features *particular* to the object of interest.

By factoring the problem into its constituent parts, we arrive at a familiar statistical domain; namely that of maximum likelihood estimation (MLE). This follows from our interest in estimating the distribution over object locations, in a sequence of images, given the initial location from whence our tracking commenced. Formally, given a sequence of images $\mathbf{x}_{1:T} \in \mathbb{R}^{H \times W \times 3}$ and an initial location for the tracked object given by a bounding box $\mathbf{b}_1 \in \mathbb{R}^4$, the conditional probability distribution factorises as

$$p(\mathbf{b}_{2:T} \mid \mathbf{x}_{1:T}, \mathbf{b}_1) = \quad = \int p(\boldsymbol{h}_1 \mid \mathbf{x}_1, \mathbf{b}_1) \prod_{t=2}^{T} \int p(\mathbf{b}_t \mid \boldsymbol{h}_t) p(\boldsymbol{h}_t \mid \mathbf{x}_t, \mathbf{b}_{t-1}, \boldsymbol{h}_{t-1}) \, \mathrm{d}\boldsymbol{h}_t \, \mathrm{d}\boldsymbol{h}_1, \quad (1)$$

where we assume that motion of an object can be described by a Markovian state $\boldsymbol{h}_t$. Our bounding box estimates are given by $\widehat{\mathbf{b}}_{2:T}$, found by the MLE of the model parameters. In sum, our contributions are threefold: Firstly, a hierarchy of attention mechanisms that leads to suppressing distractors and computational efficiency is introduced. Secondly, a biologically plausible combination of attention mechanisms and recurrent neural networks is presented for object tracking. Finally, our attention-based tracker is demonstrated using real-world sequences in challenging scenarios where previous recurrent attentive trackers have failed.

Next we briefly review related work before describing how information flows through the components of our hierarchical attention in Section 3. Section 3 details the losses applied to guide attention. Section 5 presents experiments on KTH, KITTI and ImageNet video datasets with comparison to related neural network based trackers. Section 6 discusses the results and intriguing properties of our framework and Section 7 concludes the work. Code and results are available online[1].

[1]The URL to code will be included in the published version.

2 Related Work

A number of recent studies have demonstrated that visual content can be captured through a sequence of spatial glimpses or foveation [21, 11]. Such a paradigm has the intriguing property that the computational complexity is proportional to the number of steps as opposed to the image size. Furthermore, the fovea centralis in the retina of primates is structured with maximum visual acuity in the centre and decaying resolution towards the periphery, Cheung et al. [5] show that if spatial attention is capable of zooming, a regular grid sampling is sufficient. Jaderberg et al. [13] introduced the spatial transformer network (STN) which provides a fully differentiable means of transforming feature maps, conditioned on the input itself. Eslami et al. [7] use the STN as a form of attention in combination with a recurrent neural network (RNN) to sequentially locate and identify objects in an image. Moreover, Eslami et al. [7] use a latent variable to estimate the presence of additional objects, allowing the RNN to adapt the number of time-steps based on the input. Our spatial attention mechanism is based on the two dimensional Gaussian grid filters of [15] which is both fully differentiable and more biologically plausible than the STN.

Whilst focusing on a specific location has its merits, focusing on particular appearance features might be as important. A policy with feedback connections can learn to adjust filters of a convolutional neural network (CNN), thereby adapting them to features present in the current image and improving accuracy [24]. Brabandere et al. [3] introduced dynamic filter network (DFN), where filters for a CNN are computed on-the-fly conditioned on input features, which can reduce model size without performance loss. Karl et al. [16] showed that an input-dependent state transitions can be helpful for learning latent Markovian state-space system. While not the focus of this work, we follow this concept in estimating the expected appearance of the tracked object.

In the context of single object tracking, both attention mechanisms and RNNs appear to be perfectly suited, yet their success has mostly been limited to simple monochromatic sequences with plain backgrounds [15]. Cheung [4] applied STNs [13] as attention mechanisms for real-world object tracking, but failed due to exploding gradients potentially arising from the difficulty of the data. Ning et al. [22] achieved competitive performance by using features from an object detector as inputs to a long-short memory network (LSTM), but requires processing of the whole image at each time-step. Two recent state-of-the-art trackers employ convolutional siamese networks which can be seen as an RNN unrolled over two time-steps [12, 27]. Both methods explicitly process small search areas around the previous target position to produce a bounding box offset [12] or a correlation response map with the maximum corresponding to the target position [27]. We acknowledge the recent work[2] of Gordon et al. [9] which employ an RNN based model and use explicit cropping and warping as a form of non-differentiable spatial attention. The work presented in this paper is closest to [15] where we share a similar spatial attention mechanism which is guided through an RNN to effectively learn a motion model that spans multiple time-steps. The next section describes our additional attention mechanisms in relation to their biological counterparts.

3 Hierarchical Attention

Inspired by the architecture of the human visual cortex, we structure our system around working memory responsible for storing the motion pattern and an appearance description of the tracked object. If both quantities are known, it would be possible to compute the expected location of the object at the next time step. Given a new frame, however, it is not immediately apparent which visual features correspond to the appearance description. If we were to pass them on to an RNN, it would have to implicitly solve a data association problem. As it is non-trivial, we prefer to model it explicitly by outsourcing the computation to a separate processing stream conditioned on the expected appearance. This results in a location-map, making it possible to neglect features inconsistent with our memory of the tracked object. We now proceed with describing the information flow in our model.

Given attention parameters $\mathbf{a}_t$, the *spatial attention* module extracts a glimpse $\mathbf{g}_t$ from the input image $\mathbf{x}_t$. We then apply *appearance attention*, parametrised by appearance $\boldsymbol{\alpha}_t$ and comprised of V1 and dorsal and ventral streams, to obtain object-specific features $\mathbf{v}_t$, which are used to update the hidden state $\boldsymbol{h}_t$ of an LSTM. The LSTM's output is then decoded to predict both spatial and appearance attention parameters for the next time-step along with a bounding box correction $\Delta\widehat{\mathbf{b}}_t$ for

[2][9] only became available at the time of submitting this paper.

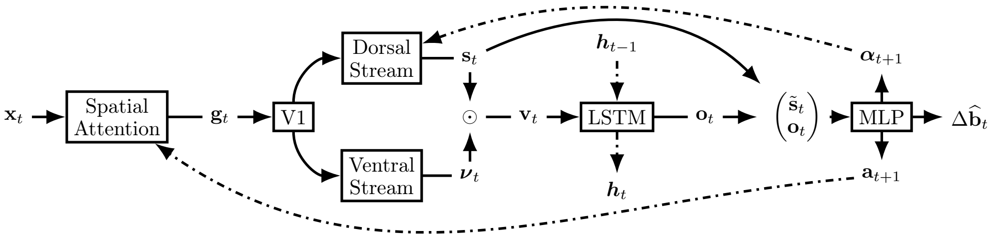


Figure 2: Hierarchical Attentive Recurrent Tracking Framework. Spatial attention extracts a glimpse $\mathbf{g}_t$ from the input image $\mathbf{x}_t$. V1 and the ventral stream extract appearance-based features while the dorsal stream computes a foreground and background segmentation of the attention glimpse $\mathbf{s}_t$. Masked features $\mathbf{v}_t$ contribute to the working memory $\boldsymbol{h}_t$. The LSTM output $\mathbf{o}_t$ is then used to compute attention $\mathbf{a}_{t+1}$, appearance $\boldsymbol{\alpha}_{t+1}$ and a bounding box correction $\Delta\widehat{\mathbf{b}}_t$. Dashed lines correspond to temporal connections, while solid lines describe information flow within one time-step.

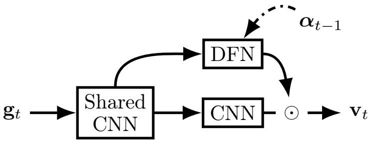


Figure 3: Architecture of the appearance attention. V1 is implemented as a CNN shared among the dorsal stream (DFN) and the ventral stream (CNN). The $\odot$ symbol represents the Hadamard product and implements masking of visual features by a locaiton map.

the current time-step. Spatial attention is driven by top-down signal $\mathbf{a}_t$, while appearance attention depends on top-down $\boldsymbol{\alpha}_t$ as well as bottom-up (contents of the glimpse $\mathbf{g}_t$) signals. Bottom-up signals have local influence and depend on stimulus salience at a given location, while top-down signals incorporate global context into local processing. This attention hierarchy, further enhanced by recurrent connections, mimics that of the human visual cortex [26]. We now describe the invdividual components of the system.

Spatial Attention Our spatial attention mechanism is similar to the one used by Kahou et al. [15]. Given an input image $\mathbf{x}_t \in \mathbb{R}^{H \times W}$, it creates two matrices $\mathbf{A}_t^x \in \mathbb{R}^{w \times W}$ and $\mathbf{A}_t^y \in \mathbb{R}^{h \times H}$, respectively. Each matrix contains one Gaussian per row; the width and positions of the Gaussians determine which parts of the image are extracted as the attention glimpse. Formally, the glimpse $\mathbf{g}_t \in \mathbb{R}^{h \times w}$ is defined as

$$\mathbf{g}_t = \mathbf{A}_t^y \mathbf{x}_t \left(\mathbf{A}_t^x\right)^{\mathsf{T}}. \tag{2}$$

Attention is described by centres μ of the Gaussians, their variances σ^2 and strides γ between centers of Gaussians of consecutive rows of the matrix, one for each axis. In contrast to the work by Kahou et al. [15], only centres and strides are estimated from the hidden state of the LSTM, while the variance depends solely on the stride. This prevents excessive aliasing during training caused when predicting a small variance (w. r. t. strides) leading to smoother convergence. The relationship between variance and stride is approximated using linear regression with polynomial basis functions (up to 4^{th} order) before training the whole system. The glimpse size we use depends on the experiment.

Appearance Attention This stage transforms the attention glimpse $\mathbf{g}_t$ into a fixed-dimensional vector $\mathbf{v}_t$ comprising appearance and spatial information about the tracked object. Its architecture depends on the experiment. In general, however, we implement $\mathrm{V1} : \mathbb{R}^{h \times w} \rightarrow \mathbb{R}^{h_v \times w_v \times c_v}$ as a number of convolutional and max-pooling layers. They are shared among later processing stages, which corresponds to the primary visual cortex in humans [6]. Processing then splits into ventral and dorsal streams. The ventral stream is implemented as a CNN, and handles visual features and outputs feature maps $\boldsymbol{\nu}_t$. The dorsal stream, implemented as a DFN, is responsible for handling spatial relationships. Let $\mathrm{MLP}(\cdot)$ denote a multi-layered perceptron. The dorsal stream uses appearance $\boldsymbol{\alpha}_t$ to dynamically compute convolutional filters $\psi_t^{a \times b \times c}$, where the superscript denotes the size of the filters and the number of feature maps, as

$$\boldsymbol{\Psi}_t = \left\{ \boldsymbol{\psi}_t^{h_i \times b_i \times c_i} \right\}_{i=1}^{K} = \mathrm{MLP}(\boldsymbol{\alpha}_t). \tag{3}$$

The filters with corresponding nonlinearities form K convolutional layers applied to the output of V1. Finally, a convolutional layer with a 1×1 kernel and a sigmoid non-linearity is applied to transform the output into a spatial Bernoulli distribution $\mathbf{s}_t$. Each value in $\mathbf{s}_t$ represents the probability of the tracked object occupying the corresponding location.

The location map of the dorsal stream is combined with appearance-based features extracted by the ventral stream, to imitate the distractor-suppressing behaviour of the human brain. It also prevents drift and allows occlusion handling, since object appearance is not overwritten in the hidden state when input does not contain features particular to the tracked object. Outputs of both streams are combined as[3]

$$\mathbf{v}_t = \mathrm{MLP}(\mathrm{vec}(\boldsymbol{\nu}_t \odot \mathbf{s}_t)), \tag{4}$$

with $\odot$ being the Hadamard product.

State Estimation Our approach relies upon being able to predict future object appearance and location, and therefore it heavily depends upon state estimation. We use an LSTM, which can learn to trade-off spatio-temporal and appearance information in a data-driven fashion. It acts like a working memory, enabling the system to be robust to occlusions and oscillating object appearance e. g., when an object rotates and comes back to the original orientation.

$$\mathbf{o}_t, \boldsymbol{h}_t = \mathrm{LSTM}(\mathbf{v}_t, \boldsymbol{h}_{t-1}), \tag{5}$$

$$\boldsymbol{\alpha}_{t+1}, \Delta\mathbf{a}_{t+1}, \Delta\widehat{\mathbf{b}}_t = \mathrm{MLP}(\mathbf{o}_t, \mathrm{vec}(\mathbf{s}_t)), \tag{6}$$

$$\mathbf{a}_{t+1} = \mathbf{a}_t + \tanh(\boldsymbol{c})\Delta\mathbf{a}_{t+1}, \tag{7}$$

$$\widehat{\mathbf{b}}_t = \mathbf{a}_t + \Delta\widehat{\mathbf{b}}_t \tag{8}$$

Equations (5) to (8) detail the state updates. Spatial attention at time t is formed as a cumulative sum of attention updates from times $t = 1$ to $t = T$, where $\boldsymbol{c}$ is a learnable parameter initialised to a small value to constrain the size of the updates early in training. Since the spatial-attention mechanism is trained to predict where the object is going to go (section 4), the bounding box $\widehat{\mathbf{b}}_t$ is estimated relative to attention at time t.

4 Loss

We train our system by minimising a loss function comprised of a: tracking loss term, a set of terms for auxiliary tasks and regularisation terms. Auxiliary tasks are essential for real-world data, since convergence does not occur without them. They also speed up learning and lead to better performance for simpler datasets. Unlike the auxiliary tasks used by Jaderberg et al. [14], ours are relevant for our main objective — object tracking. In order to limit the number of hyperparameters, we automatically learn loss weighting. The loss $\mathcal{L}(\cdot)$ is given by

$$\mathcal{L}_{\mathrm{HART}}(\mathcal{D}, \theta) = \lambda_{\mathrm{t}}\mathcal{L}_{\mathrm{t}}(\mathcal{D}, \theta) + \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{s}}(\mathcal{D}, \theta) + \lambda_{\mathrm{a}}\mathcal{L}_{\mathrm{a}}(\mathcal{D}, \theta) + R(\boldsymbol{\lambda}) + \beta R(\mathcal{D}, \theta), \tag{9}$$

with dataset $\mathcal{D} = \left\{ (\mathbf{x}_{1:T}, \mathbf{b}_{1:T})^i \right\}_{i=1}^{M}$, network parameters θ, regularisation terms $R(\cdot)$, adaptive weights $\boldsymbol{\lambda} = \{\lambda_{\mathrm{t}}, \lambda_{\mathrm{s}}, \lambda_{\mathrm{d}}\}$ and a regularisation weight β. We now present and justify components of our loss, where expectations $\mathbb{E}[\cdot]$ are evaluated as an empirical mean over a minibatch of samples $\{\mathbf{x}^i_{1:T}, \mathbf{b}^i_{1:T}\}_{i=1}^{M}$, where M is the batch size.

Tracking To achieve the main tracking objective (localising the object in the current frame), we base the first loss term on Intersection-over-Union (IoU) of the predicted bounding box w. r. t. the ground truth, where the IoU of two bounding boxes is defined as $\mathrm{IoU}(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a} \cap \boldsymbol{b}}{\boldsymbol{a} \cup \boldsymbol{b}} = \frac{\text{area of overlap}}{\text{area of union}}$. The IoU is invariant to object and image scale, making it a suitable proxy for measuring the quality of localisation. Even though it (or an exponential thereof) does not correspond to any probability

[3] vec : $\mathbb{R}^{m \times n} \to \mathbb{R}^{mn}$ is the vecorisation operator, which stacks columns of a matrix into a column vector.

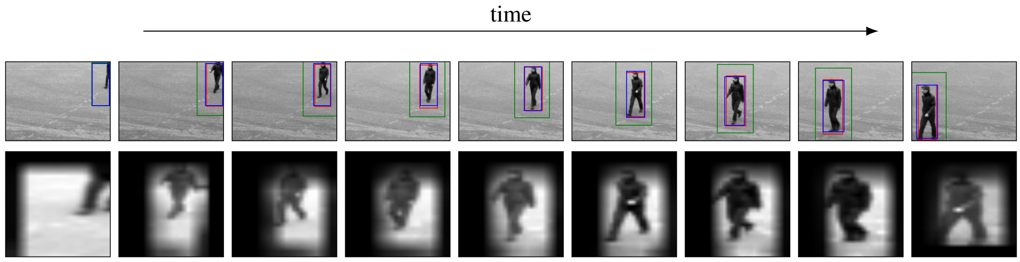


Figure 4: Tracking results on KTH dataset [23]. Starting with the first initialisation frame where all three boxes overlap exactly, time flows from left to right showing every 16^{th} frame of the sequence captured at 25fps. The colour coding follows from Figure 1. The second row shows attention glimpses multiplied with appearance attention.

distribution (as it cannot be normalised), it is often used for evaluation [18]. We follow the work by Yu et al. [28] and express the loss term as the negative log of IoU:

$$\mathcal{L}_{\mathrm{t}}(\mathcal{D},\theta) = \mathbb{E}_{p\left(\widehat{\mathbf{b}}_{1:T} \mid \mathbf{x}_{1:T},\mathbf{b}_1\right)}\left[-\log \mathrm{IoU}(\widehat{\mathbf{b}}_t,\mathbf{b}_t)\right], \tag{10}$$

with IoU clipped for numerical stability.

Spatial Attention Spatial attention singles out the tracked object from the image. To estimate its parameters, the system has to predict the object's motion. In case of an error, especially when the attention glimpse does not contain the tracked object, it is difficult to recover. As the probability of such an event increases with decreasing size of the glimpse, we employ two loss terms. The first one constrains the predicted attention to cover the bounding box, while the second one prevents it from becoming too large, with logarithmic arguments are appropriately clipped to avoid numerical instabilities:

$$\mathcal{L}_{\mathrm{s}}(\mathcal{D},\theta) = \mathbb{E}_{p(\mathbf{a}_{1:T} \mid \mathbf{x}_{1:T},\mathbf{b}_1)}\left[-\log\left(\frac{\mathbf{a}_t \cap \mathbf{b}_t}{\mathrm{area}(\mathbf{b}_t)}\right) - \log(1-\mathrm{IoU}(\mathbf{a}_t,\mathbf{x}_t))\right]. \tag{11}$$

Appearance Attention The purpose of appearance attention is to suppress distractors while keeping full view of the tracked object e. g., focus on a *particular* pedestrian moving within a group. To guide this behaviour, we put a loss on appearance attention that encourages picking out only the tracked object. Let $\tau(\mathbf{a}_t,\mathbf{b}_t) : \mathbb{R}^4 \times \mathbb{R}^4 \to \{0,1\}^{h_v \times w_v}$ be a target function. Given the bounding box $\mathbf{b}$ and attention $\mathbf{a}$, it outputs a binary mask of the same size as the output of V1. The mask corresponds to the the glimpse $\mathbf{g}$, with the value equal to one at every location where the bounding box overlaps with the glimpse and equal to zero otherwise. If we take $H(p,q) = -\sum_z p(z)\log q(z)$ to be the cross-entropy, the loss reads

$$\mathcal{L}_{\mathrm{a}}(\mathcal{D},\theta) = \mathbb{E}_{p(\mathbf{a}_{1:T},\mathbf{s}_{1:T} \mid \mathbf{x}_{1:T},\mathbf{b}_1)}[H(\tau(\mathbf{a}_t,\mathbf{b}_t),\mathbf{s}_t)]. \tag{12}$$

Regularisation We apply the L2 regularisation to the model parameters θ and to the expected value of dynamic parameters $\boldsymbol{\psi}_t(\boldsymbol{\alpha}_t)$ as $R(\mathcal{D},\theta) = \frac{1}{2}\|\theta\|_2^2 + \frac{1}{2}\left\|\mathbb{E}_{p(\boldsymbol{\alpha}_{1:T} \mid \mathbf{x}_{1:T},\mathbf{b}_1)}[\boldsymbol{\Psi}_t \mid \boldsymbol{\alpha}_t]\right\|_2^2$.

Adaptive Loss Weights To avoid hyper-parameter tuning, we follow the work by Kendall et al. [17] and learn the loss weighting $\boldsymbol{\lambda}$. After initialising the weights with a vector of ones, we add the following regularisation term to the loss function: $R(\boldsymbol{\lambda}) = -\sum_i \log(\boldsymbol{\lambda}_i^{-1})$.

5 Experiments

5.1 KTH Pedestrian Tracking

Kahou et al. [15] performed a pedestrian tracking experiment on the KTH activity recognition dataset [23] as a real-world case-study. We replicate this experiment for comparison. We use code provided by the authors for data preparation and we also use their pre-trained feature extractor. Unlike them, we did not need to upscale ground-truth bounding boxes by a factor of 1.5 and then downscale them again for evaluation. We follow the authors and set the glimpse size $(h,w) = (28,28)$. We

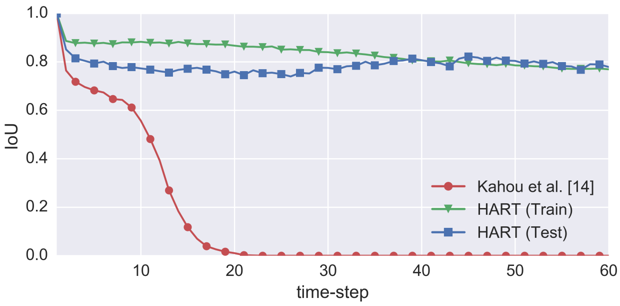


Figure 5: IoU curvers on KITTI over 60 timesteps. HART (train) present evaluation on the train set (i. e., we do not overfit).

Method	Avg. IoU
Kahou et al. [15]	0.14
Spatial Att	0.60
App Att	0.78
HART	**0.81**

Table 1: Average IoU on KITTI over 60 time-steps.

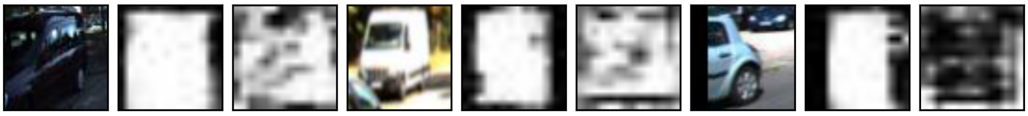
replicate the training procedure exactly, with the exception of using the RMSProp optimiser [25] with learning rate of 3.33×10^{-5} and momentum set to 0.9 instead of the stochastic gradient descent with momentum. The original work reported an IoU of 55.03% on average, on test data, while the presented work achieves an average IoU score of 77.11%, reducing the relative error by almost a factor of two. Figure 4 presents qualitative results.

5.2 Scaling to Real-World Data: KITTI

Since we demonstrated that pedestrian tracking is feasible using the proposed architecture, we proceed to evaluate our model in a more challenging multi-class scenario on the KITTI dataset [8]. It consists of 21 high resolution video sequences with multiple instances of the same class posing as potential distractors. We split all sequences into 80/20 sequences for train and test sets, respectively. As images in this dataset are much more varied, we implement V1 as the first three convolutional layers of a modified AlexNet [19]. The original AlexNet takes inputs of size 227×227 and downsizes them to 14×14 after *conv3* layer. Since too low resolution would result in low tracking performance, and we did not want to upsample the extracted glimpse, we decided to replace the initial stride of four with one and to skip one of the max-pooling operations to conserve spatial dimensions. This way, our feature map has the size of $14 \times 14 \times 384$ with the input glimpse of size $(h, w) = (56, 56)$. We apply dropout with probability 0.25 at the end of V1. The ventral stream is comprised of a single convolutional layer with a 1×1 kernel and five output feature maps. The dorsal stream has two dynamic filter layers with kernels of size 1×1 and 3×3, respectively and five feature maps each. We used 100 hidden units in the RNN with orthogonal initialisation and Zoneout [20] with probability set to 0.05. The system was trained via curriculum learning [2], by starting with sequences of length five and increasing sequence length every 13 epochs, with epoch length decreasing with increasing sequence length. We used the same optimisation settings, with the exception of the learning rate, which we set to 3.33×10^{-6}.

Table 1 and Figure 5 contain results of different variants of our model and of the RATM tracker by Kahou et al. [15] related works. *Spatial Att* does not use appearance attention, nor loss on attention parameters. *App Att* does not apply any loss on appearance attention, while *HART* uses all described modules; it is also our biggest model with 1.8 million parameters. Qualitative results in the form of a video with bounding boxes and attention are available online[4]. We implemented the RATM tracker of Kahou et al. [15] and trained with the same hyperparameters as our framework, since both are closely related. It failed to learn even with the initial curriculum of five time-steps, as RATM cannot integrate the frame $\mathbf{x}_t$ into the estimate of $\mathbf{b}_t$ (it predicts location at the next time-step). Furthermore, it uses feature-space distance between ground-truth and predicted attention glimpses as the error measure, which is insufficient on a dataset with rich backgrounds. It did better when we initialised its feature extractor with weights of our trained model but, despite passing a few stags of the curriculum, it achieved very poor final performance.

(a) The model with appearance attention loss (top) learns to focus on the tracked object, which prevents an ID swap when a pedestrian is occluded by another one (bottom).



(b) Three examples of glimpses and locations maps for a model with and without appearance loss (left to right). Attention loss forces the appearance attention to pick out only the tracked object, thereby suppressing distractors.

Figure 6: Glimpses and corresponding location maps for models trained with and without appearance loss. The appearance loss encourages the model to learn foreground/background segmentation of the input glimpse.

6 Discussion

The experiments in the previous section show that it is possible to track real-world objects with a recurrent attentive tracker. While similar to the tracker by Kahou et al. [15], our approach uses additional building blocks, specifically: (i) bounding-box regression loss, (ii) loss on spatial attention, (iii) appearance attention with an additional loss term, and (iv) combines all of these in a unified approach. We now discuss properties of these modules.

Spatial Attention Loss prevents Vanishing Gradients Our early experiments suggest that using only the tracking loss causes an instance of the vanishing gradient problem. Early in training, the system is not able to estimate object's motion correctly, leading to cases where the extracted glimpse does not contain the tracked object or contains only a part thereof. In such cases, the supervisory signal is only weakly correlated with the model's input, which prevents learning. Even when the object is contained within the glimpse, the gradient path from the loss function is rather long, since any teaching signal has to pass to the previous timestep through the feature extractor stage. Penalising attention parameters directly seems to solve this issue.

Is Appearance Attention Loss Necessary? Given enough data and sufficiently high model capacity, appearance attention should be able to filter out irrelevant input features before updating the working memory. In general, however, this behaviour can be achieved faster if the model is constrained to do so by using an appropriate loss. Figure 6 shows examples of glimpses and corresponding location maps for a model with and without loss on the appearance attention. In fig. 6a the model with loss on appearance attention is able to track a pedestrian even after it was occluded by another human. Figure 6b shows that, when not penalised, location map might not be very object-specific and can miss the object entirely (left-most figure). By using the appearance attention loss, we not only improve results but also make the model more interpretable.

Spatial Attention Bias is Always Positive To condition the system on the object's appearance and make it independent of the starting location, we translate the initial bounding box to attention parameters, to which we add a learnable bias, and create the hidden state of LSTM from corresponding visual features. In out experiments, this bias always converged to positive values favouring attention glimpse slightly larger than the object bounding box. It suggests that, while discarding irrelevant features is desirable for object tracking, the system as a whole learns to trade off attention responsibility between the spatial and appearance based attention modules.

[4] https://youtu.be/Vvkjm0FRGSs

7 Conclusion

Inspired by the cascaded attention mechanisms found in the human visual cortex, this work presented a neural attentive recurrent tracking architecture suited for the task of object tracking. Beyond the biological inspiration, the proposed approach has a desirable computational cost and increased interpretability due to location maps, which select features essential for tracking. Furthermore, by introducing a set of auxiliary losses we are able to scale to challenging real world data, outperforming predecessor attempts and approaching state-of-the-art performance. Future research will look into extending the proposed approach to multi-object tracking, as unlike many single object tracking, the recurrent nature of the proposed tracker offer the ability to attend each object in turn.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*, 2014.

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[3] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. 2016.

[4] Brian Cheung. Neural attention for object tracking. In *GTC*, 2017. URL `http://on-demand.gputechconf.com/gtc/2016/presentation/s6497-brian-cheung-neural-attention-for-object-tracking.pdf`.

[5] Brian Cheung, Eric Weiss, and Bruno A. Olshausen. Emergence of foveal image sampling from learning to attend in visual scenes. *CoRR*, abs/1611.09430, 2016.

[6] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience*, volume 806. MIT Press, 2001.

[7] SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances In Neural Information Processing Systems*, pages 3225–3233, 2016.

[8] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[9] Daniel Gordon, Ali Farhadi, and Dieter Fox. Re3: Real-time recurrent regression networks for object tracking. *arXiv preprint arXiv:1705.06368*, 2017.

[10] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

[11] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[12] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, pages 749–765. Springer, 2016.

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[14] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *CoRR*, abs/1611.05397, 2016.

[15] Samira Ebrahimi Kahou, Vincent Michalski, and Roland Memisevic. End-to-end representation learning for correlation filter based tracking. In *Open Review for CVPR Workshopp*, 2017.

[16] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. In *International Conference on Learning Representations*, 2017.

[17] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint*, 2017.

[18] Matej Kristan, Aleš Leonardis, Jiři Matas, and Michael Felsberg. *The Visual Object Tracking VOT2016 Challenge Results*. Springer International Publishing, 2016.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, 2012.

[20] David Krueger, Tegan Maharaj, János Kramár, Mohammad Pezeshki, Nicolas Ballas, Nan Rosemary Ke, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, Aaron Courville, et al. Zoneout: Regularizing rnns by randomly preserving hidden activations. *arXiv preprint arXiv:1606.01305*, 2016.

[21] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[22] Guanghan Ning, Zhi Zhang, Chen Huang, Zhihai He, Xiaobo Ren, and Haohong Wang. Spatially supervised recurrent convolutional neural networks for visual object tracking. *CoRR*, abs/1607.05781, 2016.

[23] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[24] Marijn F Stollenga, Jonathan Masci, Faustino Gomez, and Jürgen Schmidhuber. Deep networks with internal selective attention through feedback connections. In *Advances in Neural Information Processing Systems*, pages 3545–3553, 2014.

[25] T. Tieleman and G. Hinton. RMSprop Gradient Optimization. URL `http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`.

[26] Sabine Kastner Ungerleider and Leslie G. Mechanisms of visual attention in the human cortex. *Annual review of neuroscience*, 23(1):315–341, 2000.

[27] Jack Valmadre, Luca Bertinetto, João F Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *Computer Vision and Pattern Recognition*, 2017.

[28] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *ACM on Multimedia Conference*, pages 516–520. ACM, 2016.

IV. RESEARCH PROPOSAL

During the next year, I will develop a series of structured generative models of video sequences. I am going to start by building a generative model of a moving object as an extension to the AIR framework. As the problem is hard, we will start with a toy dataset of moving two-dimensional shapes. In the later version I will focus on moving three-dimensional shapes in the presence of camera motion. The final step is to improve the AIR model to work on images with rich background and real-world data, and then extending this modification to the generative models of moving object. Finally, we will investigate using trained models within the Dyna framework.

A. Variational Inference for Predictive Coding

Predictive coding describes a family of models for sequence prediction. If a sequence predictor has a hidden state, one can argue that this state should be updated only in case of imperfect predictions, see **??** for details. Friston, 2009 argues that this type of sequence modelling is employed in the human brain, where it explains phenomena related to learning. He also represents the view that the brain is Bayesian and that any prediction in the human brain has to be probabilistic. In this case, the model can be optimised by minimising the information-theoretic surprise and the prediction error can be generalised to Mahalanobis distance w. r. t. the predictive probability distribution. This approach has not been explored in the machine learning literature, and yet it gives rise to a family of models shaped after the VAE, but reformulated for prediction as opposed to reconstruction of the input. This formulation has several advantages, including:

Non-stationary priors A probabilistic prediction of the activations in the latent space can be used as a prior for the latent encoding at the next time-step. It maintains its properties as a regulariser while admitting higher flexibility of the approximate posterior distribution.

Self-normalisation Probabilistic predictive coding can be used for normalisation of activations of neural networks. Given that we minimise surprise as the learning criterion, and assuming Gaussian output probability distribution, we can use the statistics of the distribution to whiten latent encoding at layer l before inputting it to layer $l + 1$. It can potentially alleviate or even solve the problem of covariance shift in the encoder part of the model, therefore removing any need for explicit normalisation (e. g., batch normalisation). The validity of this argument is supported by the successful usage of neural baselines for variance reduction in score-function estimators (Mnih and Gregor, 2014). It is unclear how normalisation of the encoder will impact learning of the whole system, nor whether it is possible to devise a similar method of normalising the decoder activations.

While not revolutionary in itself, the predictive coding paradigm imposes structure on the model and constrains the optimisation problem, potentially leading to faster and more sample-efficient learning.

B. Unsupervised Learning to Track Objects

One can define an object in the image space as a patch of an image, where the correlation between pixels within the patch is strong, while the correlation between pixels inside and outside of the patch is weak. This definition, together with the penalty on the encoding length in the latent space, is in fact what makes AIR work. We can extend this definition to video sequences, where correlation between pixels in patches representing the same object at consecutive time-steps should be high under the assumption of high-enough frame rate. If video frames contain only simple objects on plain background, we can reformulate AIR for frame prediction instead of reconstruction to form a generative model of moving objects. This approach is unlikely to work with rich backgrounds or when an object constitutes only a small part of the scene, however. I believe that these issues can be addressed by background subtraction and soft visual attention, respectively.

1) Background Subtraction: Assuming that the appearance of the object is known and is given by a vector $\mathbf{v}_t$, it is possible to segment it out of the image by using a dynamic filter network (DFN; De Brabandere et al., 2016) in a very similar fashion to the dorsal stream of HART. Moreover, this segmentation model can be easily pre-trained (as proved by unpublished preliminary experiments) in an unsupervised way by cropping two overlapping patches from an image, treating one of them as the object and trying to find it within the second patch. If we assume that the initial tight bounding box is available (ground-truth, provided by an external object detector or AIR), we can easily extract the appearance vector.

2) Visual Attention: If the object is small relative to the image, background subtraction is unlikely to work due to the potentially large amount of noise in the segmentation. We can address this issue by using soft visual attention to crop the object or a small area around it from the image. This approach requires attention parameters, which can be initialised from the bounding box for the first frame or inferred from the sequence seen so far.

Given an attention glimpse $\mathbf{g}_t$ at time t and an appearance vector $\mathbf{v}_t$, we can create an object mask $\mathbf{m}_t$ and compute a masked version of the glimpse $\mathbf{g}_t^m = \mathbf{m}_t \odot \mathbf{g}_t$, where $\odot$ denotes the Hadamard product. Given two masked glimpses at times t and $t+1$, we can predict $\mathbf{g}_{t+1}^m$ from $\mathbf{g}_t^m$ by using an AIR-like model.

To drive learning, we can minimise the prediction loss of $\mathbf{g}_{t+1}^m$ while maximising the area of the object mask $\mathbf{m}_t$ in the image space at the same time. The trade-off here is that $\mathbf{g}_t^m$ represents the appearance of the object and only the object, therefore it cannot be used to predict anything but the object at the next time-step; therefore minimising the prediction loss will also minimise the positive area of the object mask. Maximising the object mask area will prevent it from shrinking to zero. It will also, together with minimisation of the prediction loss, encourage accurate prediction of attention parameters, since if the attention glimpse does not contain the object it is virtually impossible to predict anything in or segment that glimpse.

By combining all of the above, we arrive at a generative model of a moving object, which includes the motion model as well as the appearance model, both conditioned on the image background. Since the model is generative, we can condition it on a short video sequence and generate multiple trajectories in the image space by sampling from the prior; we can therefore examine the model for visual fidelity as well as physical plausibility of the generated paths. One caveat here is that the model will generate only the moving object, without its background. It might be necessary to use an additional component for background prediction, possibly conditioned on the predicted objects.

C. *Model-based Reinforcement Learning*

Predicting the next time-step in a structured manner might prove to be an effective way of representation learning. If a model learns intuitive physics directly from data, it can be useful model-based reinforcement learning. It remains to ask whether we can make the learning of the model faster, more general or more efficient by coupling it with an agent which can interact with its environment. Specifically, I would like to investigate the following:

1) Is it possible to use a policy for surprise-maximisation in a predictive coding setting? Can it lead to faster learning by exposing the model to otherwise rare events?
2) Can a surprise-minimisation policy be used in the absence of any explicit goal? How to avoid degenerate solutions in this case?
3) Does predicting the next time-step lead to faster learning in RL, especially when rewards are sparse? Does it encourage or discourage exploration?

Approaches described in sections IV-A and IV-B can be used for model learning in a model-based reinforcement learning setting. They can be used for pretraining or as unsupervised auxiliary tasks. While interactions between proposed methods and reinforcement learning remain unknown, they can potentially improve sample efficiency and scalability of reinforcement learning approaches and I am interested in exploring this topic.

V. CONCLUSIONS

This paper summarises the contributions I have made during my DPhil studies so far and details my future research plan. For the remainder of my studies I would like to explore representation learning for videos, with the focus on next-frame prediction by using models that impose a problem-specific structure. Finally, I would like to investigate the applicability of proposed solutions for model learning for model-based reinforcement learning.

REFERENCES

A. Krizhevsky, I. Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *NIPS*, pp. 1097–1105.

Battaglia, Peter et al. (2016). "Interaction Networks for Learning about Objects, Relations and Physics". In: *Nips*, pp. 4502–4510. arXiv: 1612.00222.

Bayer, Justin and Christian Osendorfer (2015). "Learning Stochastic Recurrent Networks". In: *ICLR*. arXiv: 1411.7610.

Bertinetto, Luca et al. (2016). "Fully-Convolutional Siamese Networks for Object Tracking". In: *ArXiv*. Springer, pp. 850–865. arXiv: 1606.09549.

Bishop, Christopher M. (2006). *Pattern recognition and machine learning*. Springer, p. 738.

Canziani, Alfredo and Eugenio Culurciello (2017). "CortexNet: a Generic Network Family for Robust Visual Temporal Representations". In: arXiv: 1706.02735.

De Brabandere, Bert et al. (2016). "Dynamic Filter Networks". In: *NIPS*. arXiv: 1605.09673.

Decety, Jean and Thierry Chaminade (2003). "When the self represents the other: A new cognitive neuroscience view on psychological identification". In: *Consciousness and Cognition*. Vol. 12. 4, pp. 577–596.

Denil, Misha et al. (2013). "Predicting Parameters in Deep Learning". In: *NIPS*, pp. 2148–2156. arXiv: 1306.0543.

Eslami, S. M. Ali et al. (2016). "Attend, Infer, Repeat: Fast Scene Understanding with Generative Models". In: *NIPS*. arXiv: 1603.08575.

Fabius, Otto and Joost R. van Amersfoort (2015). "Variational Recurrent Auto-Encoders". In: *Iclr 2013*, pp. 1–5. arXiv: 1412.6581.

Fortunato, Meire, Charles Blundell, and Oriol Vinyals (2017). "Bayesian Recurrent Neural Networks". In: arXiv: 1704.02798.

Friston, Karl (2009). "The free-energy principle: a rough guide to the brain?" In: *Trends in Cognitive Sciences* 13.7, pp. 293–301.

Häusser, Philip, Alexander Mordvintsev, and Daniel Cremers (2017). "Learning by Association - A versatile semi-supervised training method for neural networks". In: *CVPR*. arXiv: `1706.00909`.

Heess, Nicolas et al. (2017). "Emergence of Locomotion Behaviours in Rich Environments". In: arXiv: `1707.02286`.

Held, David, Sebastian Thrun, and Silvio Savarese (2016). "Learning to track at 100 FPS with deep regression networks". In: *European Conference on Computer Vision Workshop*. Vol. 9905 LNCS. Springer, pp. 749–765. arXiv: `1604.01802`.

Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh (2006). "A Fast Learning Algorithm for Deep Belief Nets". In: *Neural Computation* 18.7, pp. 1527–1554.

Kalman, R E (1960). "New Approach to Linear Filtering and Prediction Problems". In: *Fluids Engineering* 82.82 (Series D), 35–45 (1960) (11 pages).

Karl, Maximilian et al. (2017). "Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data". In: *ICLR*. arXiv: `1605.06432`.

Kastner, Sabine and Leslie G. Ungerleider (2000). "Mechanisms of visual attention in the human cortex". In: *Annual Reviews of Neuroscience* 23.1, pp. 315–341.

Kingma, Diederik P and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *ICLR*. arXiv: `1312.6114`.

Längkvist, Martin, Lars Karlsson, and Amy Loutfi (2014). "A review of unsupervised feature learning and deep learning for time-series modeling". In: *Pattern Recognition Letters* 42, pp. 11–24.

Lapedes, AS and RM Farber (1988). "How neural nets work". In: *NIPS*.

Lotter, William, Gabriel Kreiman, and David Cox (2016). "Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning". In: arXiv: `1605.08104`.

Mnih, Andriy and Karol Gregor (2014). "Neural Variational Inference and Learning in Belief Networks". In: *ICML*. arXiv: `arXiv:1402.0030v2`.

Ning, Guanghan et al. (2016). "Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking". In: *arXiv preprint arXiv:1607.05781*. arXiv: `1607.05781`.

Ondruska, Peter and Ingmar Posner (2016). "Deep Tracking: Seeing Beyond Seeing Using Recurrent Neural Networks". In: *AAAI*, pp. 3361–3367. arXiv: `1602.00991`.

Pan, Sinno Jialin and Qiang Yang (2010). *A survey on transfer learning*. arXiv: `PAI`.

Rezende, Danilo J, Shakir Mohamed, and Daan Wierstra (2014). "Stochastic backpropagation and approximate inference in deep generative models". In: *ICML*. Vol. 32, pp. 1278–1286. arXiv: `arXiv:1401.4082v3`.

Rezende, Danilo Jimenez et al. (2016). "Unsupervised Learning of 3D Structure from Images". In: *NIPS*.

Sutton, Richard S. (1991). "Dyna, an integrated architecture for learning, planning, and reacting". In: *ACM SIGART Bulletin* 2.4, pp. 160–163. arXiv: `arXiv:1011.1669v3`.

Wang, Jack M., David J. Fleet, and Aaron Hertzmann (2008). "Gaussian process dynamical models for human motion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2, pp. 283–298.