Transfer of Status Report
Generative Sequence Modelling for Reinforcement Learning

Adam Kosiorek[1]

Supervisor: Prof. Ingmar Posner

I. Introduction

Reinforcement learning (RL) allows to learn through the interaction with the environment: an agent controlled by a machine learning (ML) algorithm interacts with the world and develops a policy so as to maximise a reward. Traditional approaches to RL employed hand-designed state-spaces and tabular representations of policies, often based on state-visitation frequencies. While useful in theory, this approach is infeasible for complex real-world problems in rich environments. On one hand, designing state-space by hand is difficult as it is not clear what features are important for a particular task or type of the environment. On the other hand, the state-space is either uncountable or too big to enumerate explicitly. Model-free deep RL solves these issues by the means of function approximation with neural networks. It can learn representations from sensory inputs e. g., images, directly, thereby eliminating any need for state-space design, but it does it at a cost of a significant decrease in sample efficiency. Model-based approaches can potentially improve sample-efficiency of RL algorithms, but they constrain the maximum achievable performance as the resulting policy can be only as good as the model. Dyna, a framework combining model-free and model-based approaches introduced by Sutton, 1991, can theoretically achieve optimal performance while using imperfect models of the environments for improved sample-efficiency. In practice, it has been hard to use non-linear function approximators within Dyna, however. Firstly, the further in future we predict, the lower the quality of the prediction due to increasing uncertainty. While it is true for both linear and non-linear models, the former can provide good uncertainty estimates, which can be used to correct the resulting bias in the predictions. Secondly, non-linear models are sample-inefficient

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

and require significant amount of training before becoming useful. Before that happens, they can destabilise training of the model-free policy by providing inaccurate predictions.

While it is hard to provide high-quality predictions in the image space, especially over long time-horizons, it is not clear whether it is necessary, or whether all parts of the image have to be predicted with equal accuracy. Consider the task of assembling an object from its parts: there are several parts lying on a workbench and the goal is to arrange them in a specific configuration. While the exact appearance of the final scene or what is in the background does not matter, absolute poses and identities of object parts as well as relations between them do. It is interesting to ask whether we can build a non-parametric unsupervised latent-variable model of the scene, where latent variables would explain objects and their poses and where the encoding length would depend on the number of objects in the scene (hence non-parametric). Moreover, is it possible to perform prediction or a model-based simulation in the latent space, so as to circumvent deteriorating prediction quality, often visible in the image space? Finally, would it be possible to use such latent-space simulations within the Dyna framework, especially with a pre-trained model of the environment dynamics? It is worth noting that decomposing a scene into its constituent parts might enforce conditional independence properties between the objects and/or the scene, making it harder to implicitly reason about existing relations. It begs asking the following question: does scene decomposition require to consider intra-object relations explicitly or is implicit treatment sufficient?

To answer the above questions, we would like to focus on the problem of unsupervised state estimation via generative modelling, and specifically on approaches that (i) can perform multiple number of computation steps per input to support the variable-length representation of the scene, (ii) allow simulation in the absence of data and define prior distributions from which samples can be drawn when data are absent, (iii) support on-line training, which is necessary for a scalable use within the Dyna framework, (iv) estimate a Markovian state, since the environment in many real-world problems, especially involving robots, is partially-observable, (v) are stochastic and thereby able to generate multiple state-space trajectories from a single starting state, which accounts for imperfect information and encourages better state-space exploration necessary for sample-efficient learning of RL models.

While there exist multiple approaches that meet the above requirements, we would like to focus on the recent advances in variational inference for neural networks. Variational Autoencoders allow building scalable generative latent-variable models of high-dimensional data, which is necessary for our task, and they have been shown to work with variable number of steps per input. In contrast to standard neural networks, they are stochastic and they provide prior distributions on the latent

representation. Unlike Gaussian Processes, they allow on-line training and at the same time the computational complexity of inference is independent of the size of the training set.

The rest of this report is structured as follows. Section II covers prior work related to the areas in question. We summarise the task of sequence prediction, describe relevant variants of unsupervised learning, investigate how model structure helps to learn abstract concepts from data and examine prior work on model-based RL. In Section III, We describe our prior work on object tracking and how it ties with our interests and the planned future work. Section IV details how we are going to build a structured generative dynamics model and use it in reinforcement learning. Section V concludes this work. We provide our prior work on object tracking as an example publication in the Appendix.

II. Related Work

We start the discussion of related works with the Dyna framework and describe the role of its components. We then proceed to the tasks of unsupervised learning and sequence modelling, which are important for incorporating prior information and learning non-stationary models of environment dynamics, respectively, which we hope can be used to pre-train a model of the environment to be used within Dyna. Finally, we touch on how *appropriate* structuring of the model can aid representation learning, thereby improving the quality of the environment model.

A. Generative Modelling for Reinforcement Learning

Model-free RL is data hungry and improving sample efficiency of model-free methods is a long-standing research problem. Sutton, 1991 introduced the Dyna architecture, which uses and jointly trains a model-free parametric policy and a generative model of the environment. The former allows efficient inference and optimal performance, the latter reduces number of samples required from the environment by providing model-based simulations. Despite the theoretical advantages, it has been very difficult in practice to implement Dyna for anything but the simplest RL problems due to instabilities introduced by the learning of non-linear function approximators. Gu et al., 2016 managed to use use the Dyna framework with a non-linear policy network, but showed that using neural networks for the environment model leads to lower performance than using linear approximators due to slow learning of the former. Nagabandi et al., 2017 has overcome this issue by using a mid-sized neural network as the environment model and pre-training it without supervision on random walks through the state-space. This work is probably the closest to the proposed approach and it shows that Dyna can work with non-linear function approximators of the environment dynamics.

The Dyna framework has neuroscientific grounding. It has been hypothesised that the parametric models of neocortex admit efficient inference but require long time to train. According to Kumaran, Hassabis, and McClelland, 2016, this issue can be mitigated by hippocampus, which can quickly store experiences and either replay or simulate them during sleep. In this sense, simulations in Dyna are very similar to the experience-replay mechanism, which has been shown to stabilise and improve convergence of large-scale model-free RL models (Mnih et al., 2015).

Dyna is not the only approach based on generative modelling. On the contrary, RL based on control in latent spaces has been quite successful. Watter et al., 2015 introduced Embed to Control (E2C), a stochastic locally-optimal control framework, which uses Variational Autoencoders (VAEs; *cf.* Section II-B) for learning of the latent-space for control. It approximates the latent-space dynamics by a locally-linear transition, which has controls as one of the inputs. The VAE manages to recover the true latent variables describing the state of the environment, which results in good long-term prediction performance. This type of long-term imagination could be used for multi-step roll-outs of model-based simulations in Dyna.

B. Unsupervised Learning via Generative Modelling

While data in general are abundant and cheap, data for supervised learning are often expensive and time-consuming to gather. The majority of ML algorithms require relatively large amounts of labelled training data. Lake et al., 2016 suggest that machine learning typically starts 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). It is possible to learn without supervision by 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 latter *explain* the former 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 $q_\phi(\mathbf{z})$ parametrised by parameters ϕ and approximate the true data distribution by maximising the evidence lower bound (ELBO) $\mathcal{L}_{VAE}(\mathbf{x})$ on the log-probability of the data $\log p(x) = \mathcal{L}_{VAE}(\mathbf{x}) + \mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid p(\mathbf{z} \mid \mathbf{x}))$, where $p(\mathbf{z} \mid \mathbf{x})$ is the true (intractable) posterior distribution. This approach results in variational autoencoders (VAE; Kingma and Welling, 2014; Rezende, Mohamed, and Wierstra, 2014). VAEs are much more flexible than DBNs as they allow latent variables from arbitrary probability distribution

functions (pdf), can be trained end-to-end with off-the-shelf gradient-based methods and, in contrast to DBNs which require Markov Chain Mote Carlo sampling, admit efficient inference. 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.

VAEs are especially interesting for the problem at hand because of their Bayesian nature: they are stochastic and they impose a prior on the latent space. ELBO can be written as $\mathcal{L}_{VAE}(\mathbf{x}) = -\mathbb{E}_{q(\mathbf{z})}[p_\theta(\mathbf{x} \mid \mathbf{z})] + \mathrm{KL}(q(\mathbf{z}) \,||\, p(\mathbf{z}))$, where the second term is the KL-divergence between the approximate posterior and the prior. It can be interpreted as an encoding-length penalty that encourages minimum-coding length, but it also forces a *particular* shape on the approximate posterior. It allows substituting samples from the prior for data, thereby creating new samples in the observation space, which is useful for creating model-based state-space roll-outs for Dyna.

C. 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; Bayer and Osendorfer, 2015; Fortunato, Blundell, and Vinyals, 2017). 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). Even though deterministic RNNs can be forced to learn low-dimensional representations by limiting the number of neurons in a layer, doing so can constrain learning and reduce performance. In the initial stages of optimisation, neural models tend to explore all the available dimensions and only later converge to informative representations (Shwartz-Ziv and Tishby, 2017). It is possible to encourage sparsity (Engelcke et al., 2016) or compress the model (Han, Mao, and Dally, 2015), but these approaches are based on heuristics. Variational methods, on the other hand, ensure compact latent representations in a principled way by directly minimising the encoding length, see Section II-B, but also Bishop, 2006 and Burda, Grosse, and Salakhutdinov, 2015. Low-dimensional latent representation is important for our problem as difficulty of controlling an agent increases with increasing number of dimensions of the control space (Watter et al., 2015). While the control space is independent of the environment model, the latter should be conditioned on control inputs and controls can then be seen as controlling the evolution of the environment in the latent space.

D. Model Structure as Prior Information

As the majority of neural models are over-parametrised (Denil et al., 2013), learning abstract notions from data can be extremely sample inefficient. Eslami et al., 2016 introduced 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. In the context of this report, a similar approach can determine parts of the scene that are relevant to an RL task; it also creates a separate latent representation for each object, thereby allowing explicit reasoning about single objects and relations between them.

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. Their application is to simulate physical systems under full observability, but additionally, the model structure enables learning invariants (e. g., energy conservation) and inferring latent variables describing the system as a whole (e. g., potential energy). Since we are interested in decomposing a scene into its parts and simulating the evolution of that scene, we can treat it as a physics system. Introducing structure that makes it easier to infer intuitive physics and system-level invariants can improve performance, especially when considering predictions under long temporal horizons. In the continuation of this work, Santoro et al., 2017 introduces *relation networks* (RN). This simple component is permutation invariant and

can process pairs of objects and infer relations between them. A major shortcoming of the approach is identifying objects. With our approach, however, explicit latent representation for every object is available by design; introducing RN as a component can be both helpful and simple.

III. Submitted Work

During my first year as a DPhil student we developed the Hierarchical Attentive Recurrent Tracking (HART) framework, which was submitted to NIPS 2017, see Appendix. This RNN-based model learns to track objects in videos by focusing on small image regions. 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, we 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).

The task of object tracking is fully-supervised, but can be seen as a reinforcement learning problem (Zhang et al., 2017) with a continuous action space, where a policy chooses a bounding-box update at every time-step; the agent receives a reward either at every time-step or at the end of the episode and the reward structure can be chosen based on the distance between the ground-truth bounding-box and the model estimate. In this setup, HART can be seen as a model-free policy. Instead of using a pre-trained feature extractor, it would be possible to utilise a model of the environment to perform off-line training of the policy, similarly to the Dyna framework. If the model is structured and provides correct position estimates of the object, this approach could increase performance of the tracking framework via unlimited model-based data augmentation.

Alternatively, if a generative latent-variable model of image sequences is available, HART could use the latent representation as extracted features, without the need to rely on a feature extractor pre-trained on static image analysis. Even though static image analysis has different characteristics than sequential analysis (e. g., data redundancy at consecutive time-steps), image classification models are often used for processing of video (see e. g., Ning et al., 2016).

Our work on HART resulted in a biologically-inspired algorithm, which advanced the state-of-the-art performance in attentive recurrent tracking. Contrary to modern trackers, it does not use heuristics to update the scale estimate of the tracked object or to choose the search region in the new frame (Bertinetto et al., 2016; Held, Thrun, and Savarese, 2016). It is efficient thanks to the attention mechanism and end-to-end trainable. Finally, it has taught us about learning in the presence of

temporal dependencies and structured modelling.

IV. Research Proposal

During the remainder of this DPhil, we will focus on answering the following question: How accurate does a model of the environment have to be in order to be useful in the context of model-based RL? Specifically, we are interested in finding out (i) whether it is possible to identify *important* parts of the scene without any supervision, (ii) if simulating their dynamics separately and explicitly brings about any benefits and (iii) how relations between them affect the simulation. To this end, we are going to develop a structured non-parametric generative latent-variable model of the environment dynamics. By imposing a specific structure within the VAE framework, we can enforce scene decomposition by putting an *appropriate* prior on the latent space, similar to AIR, and thereby gaining insight into what the model deems to be important. Making the model non-parametric allows us to work with a variable number of objects, which allows tackling scenes and tasks of varying complexity. It also enables us to simulate objects separately and analyse relations between them, thereby answering question (ii) and (iii). Using a generative model is required by model-based RL, while at the same time providing insights into its internal workings: we can qualitatively evaluate samples generated by the model. To validate our endeavours, we are going to employ our generative model in the Dyna framework and use it in simulation in partially-observable environments involving robotics and/or computer games. We now detail the above.

A. Generalisation of the Attend, Infer, Repeat framework

The AIR framework can be seen as a non-parametric latent-variable generative model of images, which is very close to what we would like to develop. It reconstructs an image by detecting objects present therein without supervision and paiting one object at a time in a blank canvas. The original work showed it to work only in the context of plain backgrounds (black, in fact) and simple objects. In order for the model to be useful in any real-world setting, it has to be able to handle images with rich backgrounds and occluded and complex objects. We expect that this will require a form of object/background segmentation or background subtraction and generative blending of objects and the background. Given that AIR uses a spatial transformer (Jaderberg et al., 2015) to draw objects in a canvas, it is straightforward to create an explanation mask, which marks which locations in the canvas has been drawn to. When objects are explained, it should be possible to use the explanation mask with a complementary background model to explain the remainder of the image. Separating reconstruction of the background and the objects might create discontinuities at the

boundaries, however, and it is unclear how to prevent the background model from explaining the objects at the same time. It is our intuition that pixel correlations within objects are different than in the background or between objects and their neighbourhoods. If we parametrise background- and object-generating models with a minimum-length encoding scheme, it should force them to learn their problem-specific correlation structure, therefore forcing the parts of the scene to be explained by corresponding model components. Since the KL-divergence term in the VAE loss can be interpreted as an information-bottleneck (Achille and Soatto, 2016), VAE effectively minimises encoding-length of the latent representation.

We will start by working with a multi-MNIST dataset, similar to the one used by the original paper, but with a noisy background. The goal is to upscale the approach to real-world images and e. g., ImageNet dataset. At this point it is unclear whether the approach based on AIR will work. If it does not, we might have to introduce more structure by e. g., image segmentation or employ semi-supervised learning instead of a purely unsupervised approach. We expect this phase to take between 4 and 6 months and result in an article submitted to either IJCAI or ICML in February 2018.

B. A Generative Dynamical Model of Moving Objects

Once we are able to detect salient parts of an image, we are going to focus on modelling environment dynamics in terms of dynamics of individual objects. We will focus on developing a transition function in the latent space, so that the reconstruction after the transition will lead to prediction of the scene at the following time-step. It will results in a model that is effectively able to track objects by generating them one-at-a-time in a sequence of blank canvases. To reconstruct an image, AIR decomposes it into a set of $\mathbf{z}^{\text{where}}$ and $\mathbf{z}^{\text{what}}$ latent variables, which describe location and appearance of an object, respectively. The sequential model will need to take time-dependencies into account. In particular, instead of directly using $\mathbf{z}^{\text{where}}$ and $\mathbf{z}^{\text{what}}$ inferred from an image $\mathbf{x}_t$ at time t to reconstruct the image at time $t+1$, it will need to take into account the history of appearances and locations $\mathbf{z}_{1:t}$ at times 1 to t. This can be accomplished by using a learned dynamics model, e. g., an RNN. In practice, the dynamics model should take into account actions executed by the agent, which might lead to transition functions similar to the ones of Watter et al., 2015 and Karl et al., 2017.

Even though the modification to AIR looks simple, it is unclear whether this approach will work. Firstly. it is based on the assumption (like AIR) that the correlation between pixels within an object is much stronger than correlation between pixels inside and outside of the object. Secondly, this

model is not allowed to peek at the image at time $t+1$ to reconstruct it, which severely increases the difficulty of the task. To address this issues, we are going to start simple, with a toy dataset of moving two-dimensional shapes. We will extend it later to moving three-dimensional shapes in the presence of camera motion.

In the absence of data, the model allows simulation by updating the latent state $\mathbf{z}_t$ with samples drawn from a prior distribution $p(\mathbf{z})$. Choosing the right prior for a sequential task poses a research question by itself (Sölch et al., 2016) and might require significant effort to answer. The transition function of the dynamics model is another crucial component of the model. It defines dynamics in the latent space and it will determine whether the model adheres to the laws of physics. We expect this stage to take about two to four months and we expect that it will lead to a publication, possibly for NIPS, whose deadline is in May 2018.

C. A Generative Model of Videos

Combining the generalised version of AIR with a generative model of moving objects will result in a structured generative model of image sequences. While simple in principle, we expect a number of issues to arise. Firstly, the moving object model does not take the background of the target image into account. We might have to modify the model to predict the background and use it to condition the locations of the objects in the target image; alternatively we can condition background generation on the object appearance and their location. Secondly, training a high-fidelity model on video sequences is computationally very expensive due to the huge amounts of data this approach requires. Additionally, it is unclear which output probability distribution to use; output probability distribution in VAEs is responsible for the shape of the loss landscape. Gaussian assumption about prediction errors is not well justified when dealing with images and temporal dependencies between model outputs aggravate this issue even further. This issue might require further research (Generative Adversarial Networks might hold an answer; Wenzhe et al., 2016) if satisfactory performance is to be attained. We expect this stage to take 4 to 6 months. We would like to turn it into a publication for ICLR with the deadline in October 2018.

D. Model-based RL

With the above components ready, we will be in a position to answer the initial question about reinforcement learning: how accurate does the model of the environment have to be? We will attempt to use our object-centric dynamic model of the environment in the Dyna framework and compare it to baselines. It will allow us to examine how does the non-parametric treatment of parts

of the scene affect simulation within the RL context and whether object-centric representation is useful for a model-free policy and whether explicit reasoning about relations between objects is of importance. By enforcing handling different parts of the scene by different model components, we will be able to examine how does the representation type of various parts of the scene affect performance of an RL agent. This approach has the potential for improving sample efficiency within the Dyna framework while granting improved interpretability. Given difficulties with training non-linear models of environment, however, it is unclear whether this approach will work. A generative model of videos with variable-length encoding factorised between parts of the scene can be very useful in latent-space control algorithms similar to E2C, especially when any form of relational reasoning is required. In this case, the object-based representation delivered by AIR-like modelling can be married with structured reasoning models such as relational nets of Santoro et al., 2017 or the dynamic neural computer of Graves et al., 2016. We expect the final stage to take the remainder of this DPhil.

V. CONCLUSIONS

This report summarises the contributions I have made during my DPhil studies so far and details my future research plan. For the remainder of my studies we would like to explore learning of structured generative latent-variable models sequential data, with the goal of using developed techniques in 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.

Achille, Alessandro and Stefano Soatto (2016). "Information Dropout: Learning Optimal Representations Through Noisy Computation". In: *arXiv*, pp. 1–11. arXiv: 1611.01353.

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.

Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov (2015). "Importance Weighted Autoencoders". In: arXiv: 1509.00519.

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.

Engelcke, Martin et al. (2016). “Vote3Deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks”. In: arXiv: 1609.06666.

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.

Graves, Alex et al. (2016). “Hybrid computing using a neural network with dynamic external memory”. In: *Nature* 538.7626, pp. 471–476.

Gu, Shixiang et al. (2016). “Continuous Deep Q-Learning with Model-based Acceleration”. In: arXiv: 1603.00748.

Han, Song, Huizi Mao, and William J. Dally (2015). “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding”. In: arXiv: 1510.00149.

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.

Jaderberg, Max et al. (2015). “Spatial Transformer Networks”. In: *Nips*, pp. 1–14. arXiv: arXiv:1506.02025v1.

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.

Kumaran, Dharshan, Demis Hassabis, and James L. McClelland (2016). “What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated”. In: *Trends in Cognitive Sciences* 20.7, pp. 512–534.

Lake, Brenden M. et al. (2016). “Building Machines that learn and think like people”. In: *arXiv:1604.00289v1[cs.* 2012, pp. 1–54. arXiv: 1604.00289.

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*.

Mnih, Volodymyr et al. (2015). “Human-level control through deep reinforcement learning”. In: *Nature* 518.7540, pp. 529–533.

Nagabandi, Anusha et al. (2017). “Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning”. In: arXiv: 1708.02596.

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

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

Rezende, Danilo Jimenez, S. M. Ali Eslami, et al. (2016). “Unsupervised Learning of 3D Structure from Images”. In: *NIPS*.

Rezende, Danilo Jimenez, 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.

Santoro, Adam et al. (2017). “A simple neural network module for relational reasoning”. In: *Arxiv*, pp. 1–16. arXiv: 1706.01427.

Shwartz-Ziv, Ravid and Naftali Tishby (2017). “Opening the Black Box of Deep Neural Networks via Information”. In: arXiv: 1703.00810.

Sölch, Maximilian et al. (2016). “Variational Inference for On-line Anomaly Detection in High-Dimensional Time Series”. In: *ICML*. arXiv: 1602.07109.

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.

Watter, Manuel et al. (2015). “Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images”. In: *NIPS*. arXiv: 1506.07365.

Wenzhe, Shi et al. (2016). “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network”. In: *CVPR*, pp. 1874–1883. arXiv: 1609.05158.

Zhang, Da et al. (2017). “Deep Reinforcement Learning for Visual Object Tracking in Videos”. In: arXiv: 1701.08936.

APPENDIX

Hierarchical Attentive Recurrent Tracking was submitted to NIPS 2017.

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.

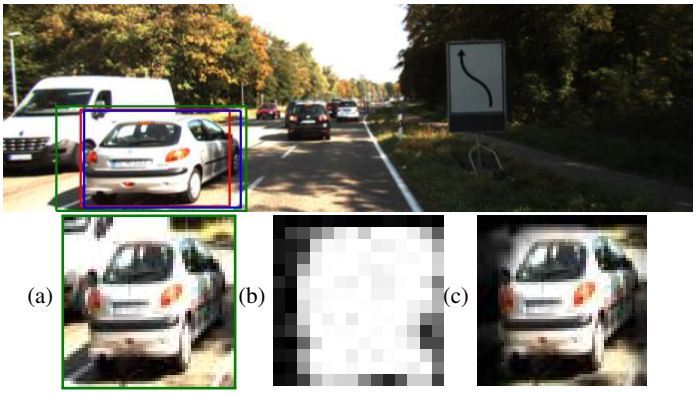

Figure 1: KITTI image with the ground-truth and predicted bounding boxes and an attention glimpse. The lower row corresponds to the hierarchical attention of our model: 1^{st} layer extracts an attention glimpse (a), the 2^{nd} layer uses appearance attention to build a location map (b). The 3^{rd} layer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

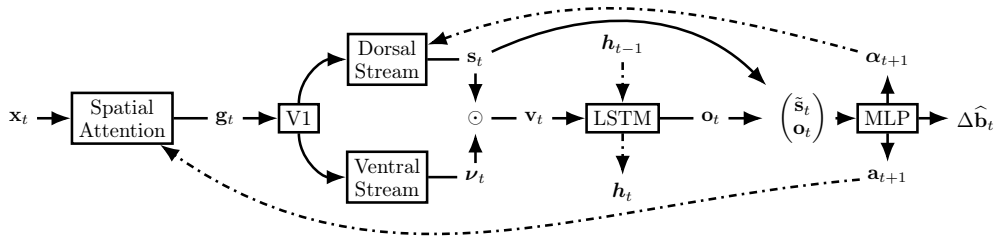

Figure 2: Hierarchical Attentive Recurrent Tracking Framework. Spatial attention extracts a glimpse $\mathbf{g}_t$ from the input image $\mathbf{x}_t$. V1 and the ventral stream extract appearance-based features while the dorsal stream computes a foreground and background segmentation of the attention 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.

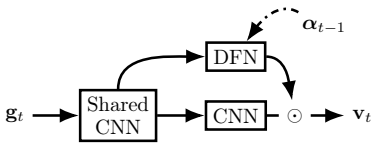

Figure 3: Architecture of the appearance attention. V1 is implemented as a CNN shared among the dorsal stream (DFN) and the ventral stream (CNN). The $\odot$ symbol represents the Hadamard product and implements masking of visual features by a locaiton map.

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 $\mathrm{V1} : \mathbb{R}^{h\times w} \to \mathbb{R}^{h_v\times w_v\times c_v}$ 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 $\boldsymbol{\psi}_t^{a\times b\times c}$, 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 $\left\{\mathbf{x}^i_{1:T}, \mathbf{b}^i_{1:T}\right\}_{i=1}^{M}$, 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] $\mathrm{vec} : \mathbb{R}^{m \times n} \to \mathbb{R}^{mn}$ is the vecorisation operator, which stacks columns of a matrix into a column vector.

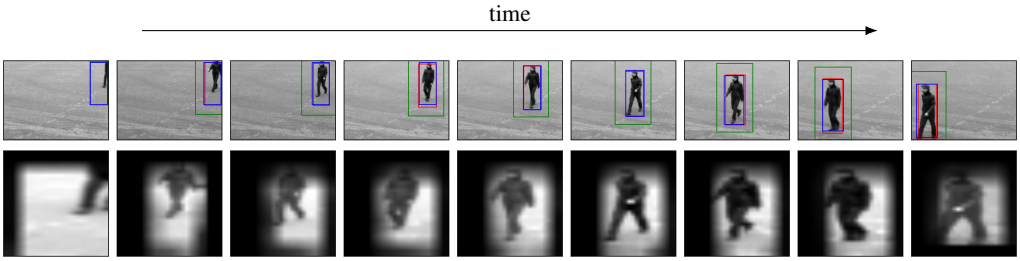

Figure 4: Tracking results on KTH dataset [23]. Starting with the first initialisation frame where all three boxes overlap exactly, time flows from left to right showing every 16^{th} frame of the sequence captured at 25fps. The colour coding follows from Figure 1. The second row shows attention glimpses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

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

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:

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

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

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

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 $R(\mathcal{D}, \theta) = \frac{1}{2}\|\theta\|_2^2 + \frac{1}{2}\big\|\mathbb{E}_{p(\boldsymbol{\alpha}_{1:T} | \mathbf{x}_{1:T}, \mathbf{b}_1)}[\boldsymbol{\Psi}_t \mid \boldsymbol{\alpha}_t]\big\|_2^2$.

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

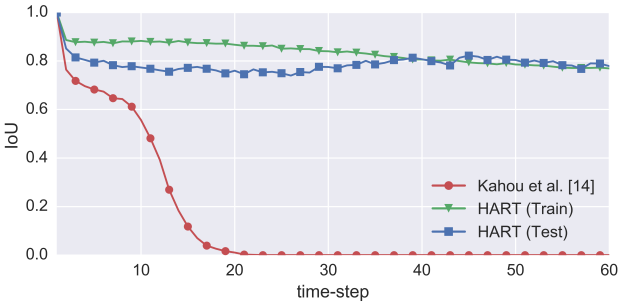

Figure 5: IoU curvers on KITTI over 60 timesteps. HART (train) present evaluation on the train set (i. e., we do not overfit).

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.