Learning Object-Centric Representations

Adam Roman Kosiorek

Wolfson College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas 2019

Abstract

Your abstract text goes here. Check your departmental regulations, but generally this should be less than 300 words. See the beginning of Chapter **??** for more.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque sit amet nibh volutpat, scelerisque nibh a, vehicula neque. Integer placerat nulla massa, et vestibulum velit dignissim id. Ut eget nisi elementum, consectetur nibh in, condimentum velit. Quisque sodales dui ut tempus mattis. Duis malesuada arcu at ligula egestas egestas. Phasellus interdum odio at sapien fringilla scelerisque. Mauris sagittis eleifend sapien, sit amet laoreet felis mollis quis. Pellentesque dui ante, finibus eget blandit sit amet, tincidunt eu neque. Vivamus rutrum dapibus ligula, ut imperdiet lectus tincidunt ac. Pellentesque ac lorem sed diam egestas lobortis.

Suspendisse leo purus, efficitur mattis urna a, maximus molestie nisl. Aenean porta semper tortor a vestibulum. Suspendisse viverra facilisis lorem, non pretium erat lacinia a. Vestibulum tempus, quam vitae placerat porta, magna risus euismod purus, in viverra lorem dui at metus. Sed ac sollicitudin nunc. In maximus ipsum nunc, placerat maximus tortor gravida varius. Suspendisse pretium, lorem at porttitor rhoncus, nulla urna condimentum tortor, sed suscipit nisi metus ac risus.

Aenean sit amet enim quis lorem tristique commodo vitae ut lorem. Duis vel tincidunt lacus. Sed massa velit, lacinia sed posuere vitae, malesuada vel ante. Praesent a rhoncus leo. Etiam sed rutrum enim. Pellentesque lobortis elementum augue, at suscipit justo malesuada at. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent rhoncus convallis ex. Etiam commodo nunc ex, non consequat diam consectetur ut. Pellentesque vitae est nec enim interdum dapibus. Donec dapibus purus ipsum, eget tincidunt ex gravida eget. Donec luctus nisi eu fringilla mollis. Donec eget lobortis diam.

Suspendisse finibus placerat dolor. Etiam ornare elementum ex ut vehicula. Donec accumsan mattis erat. Quisque cursus fringilla diam, eget placerat neque bibendum eu. Ut faucibus dui vitae dolor porta, at elementum ipsum semper. Sed ultrices dui non arcu pellentesque placerat. Etiam posuere malesuada turpis, nec malesuada tellus malesuada.

Learning Object-Centric Representations

Adam Roman Kosiorek

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2019

Acknowledgements

Personal

This is where you thank your advisor, colleagues, and family and friends.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum feugiat et est at accumsan. Praesent sed elit mattis, congue mi sed, porta ipsum. In non ullamcorper lacus. Quisque volutpat tempus ligula ac ultricies. Nam sed erat feugiat, elementum dolor sed, elementum neque. Aliquam eu iaculis est, a sollicitudin augue. Cras id lorem vel purus posuere tempor. Proin tincidunt, sapien non dictum aliquam, ex odio ornare mauris, ultrices viverra nisi magna in lacus. Fusce aliquet molestie massa, ut fringilla purus rutrum consectetur. Nam non nunc tincidunt, rutrum dui sit amet, ornare nunc. Donec cursus tortor vel odio molestie dignissim. Vivamus id mi erat. Duis porttitor diam tempor rutrum porttitor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed condimentum venenatis consectetur. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Aenean sit amet lectus nec tellus viverra ultrices vitae commodo nunc. Mauris at maximus arcu. Aliquam varius congue orci et ultrices. In non ipsum vel est scelerisque efficitur in at augue. Nullam rhoncus orci velit. Duis ultricies accumsan feugiat. Etiam consectetur ornare velit et eleifend.

Suspendisse sed enim lacinia, pharetra neque ac, ultricies urna. Phasellus sit amet cursus purus. Quisque non odio libero. Etiam iaculis odio a ex volutpat, eget pulvinar augue mollis. Mauris nibh lorem, mollis quis semper quis, consequat nec metus. Etiam dolor mi, cursus a ipsum aliquam, eleifend venenatis ipsum. Maecenas tempus, nibh eget scelerisque feugiat, leo nibh lobortis diam, id laoreet purus dolor eu mauris. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nulla eget tortor eu arcu sagittis euismod fermentum id neque. In sit amet justo ligula. Donec rutrum ex a aliquet egestas.

Institutional

If you want to separate out your thanks for funding and institutional support, I don't think there's any rule against it. Of course, you could also just remove the subsections and do one big traditional acknowledgement section.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut luctus tempor ex at pretium. Sed varius, mauris at dapibus lobortis, elit purus tempor neque,

facilisis sollicitudin felis nunc a urna. Morbi mattis ante non augue blandit pulvinar. Quisque nec euismod mauris. Nulla et tellus eu nibh auctor malesuada quis imperdiet quam. Sed eget tincidunt velit. Cras molestie sem ipsum, at faucibus quam mattis vel. Quisque vel placerat orci, id tempor urna. Vivamus mollis, neque in aliquam consequat, dui sem volutpat lorem, sit amet tempor ipsum felis eget ante. Integer lacinia nulla vitae felis vulputate, at tincidunt ligula maximus. Aenean venenatis dolor ante, euismod ultrices nibh mollis ac. Ut malesuada aliquam urna, ac interdum magna malesuada posuere.

Abstract

Your abstract text goes here. Check your departmental regulations, but generally this should be less than 300 words. See the beginning of Chapter **??** for more.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque sit amet nibh volutpat, scelerisque nibh a, vehicula neque. Integer placerat nulla massa, et vestibulum velit dignissim id. Ut eget nisi elementum, consectetur nibh in, condimentum velit. Quisque sodales dui ut tempus mattis. Duis malesuada arcu at ligula egestas egestas. Phasellus interdum odio at sapien fringilla scelerisque. Mauris sagittis eleifend sapien, sit amet laoreet felis mollis quis. Pellentesque dui ante, finibus eget blandit sit amet, tincidunt eu neque. Vivamus rutrum dapibus ligula, ut imperdiet lectus tincidunt ac. Pellentesque ac lorem sed diam egestas lobortis.

Suspendisse leo purus, efficitur mattis urna a, maximus molestie nisl. Aenean porta semper tortor a vestibulum. Suspendisse viverra facilisis lorem, non pretium erat lacinia a. Vestibulum tempus, quam vitae placerat porta, magna risus euismod purus, in viverra lorem dui at metus. Sed ac sollicitudin nunc. In maximus ipsum nunc, placerat maximus tortor gravida varius. Suspendisse pretium, lorem at porttitor rhoncus, nulla urna condimentum tortor, sed suscipit nisi metus ac risus.

Aenean sit amet enim quis lorem tristique commodo vitae ut lorem. Duis vel tincidunt lacus. Sed massa velit, lacinia sed posuere vitae, malesuada vel ante. Praesent a rhoncus leo. Etiam sed rutrum enim. Pellentesque lobortis elementum augue, at suscipit justo malesuada at. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent rhoncus convallis ex. Etiam commodo nunc ex, non consequat diam consectetur ut. Pellentesque vitae est nec enim interdum dapibus. Donec dapibus purus ipsum, eget tincidunt ex gravida eget. Donec luctus nisi eu fringilla mollis. Donec eget lobortis diam.

Suspendisse finibus placerat dolor. Etiam ornare elementum ex ut vehicula. Donec accumsan mattis erat. Quisque cursus fringilla diam, eget placerat neque bibendum eu. Ut faucibus dui vitae dolor porta, at elementum ipsum semper. Sed ultrices dui non arcu pellentesque placerat. Etiam posuere malesuada turpis, nec malesuada tellus malesuada.

Contents

List of Figures

List of Abbreviations

1-D, 2-D . . . One- or two-dimensional, referring in this thesis to spatial dimensions in an image.

Otter . . . . . One of the finest of water mammals.

Hedgehog . . . Quite a nice prickly friend.

Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit...

There is no one who loves pain itself, who seeks after it and wants to have it, simply because it is pain...

— Cicero's *de Finibus Bonorum et Malorum*

1 Introduction

Contents

1.1 Motivation

The rapid advance of minimally-invasive cardiac procedures promises improvements in patient safety, procedure efficacy, and access to treatment. While percutaneous coronary intervention (PCI) has become routine and highly effective **bravata_systematic_2007**, catheter procedures in areas such as electrophysiology (EP) and valve replacement are still coming of age. This progress is driven by demographics and the improvement in general cardiac care, as patients surviving initial cardiac events go on to require treatment for sequelae **foot_demographics_2000**. The growing need for advanced treatment is being answered by developments in

catheter technology and procedures. These tools are continually advancing to access and manipulate an ever-broader range of anatomy **sousa_new_2005**.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sagittis dolor at nulla feugiat, vitae iaculis est rutrum. Mauris eu sem eros. Sed id faucibus urna. In egestas eros et sapien egestas imperdiet. In hac habitasse platea dictumst. Phasellus vitae varius tortor. Mauris nec sollicitudin enim. Suspendisse molestie leo nec mauris molestie, nec imperdiet magna vehicula. Phasellus sodales tortor dui, a lacinia turpis congue at. Pellentesque mattis dui non libero commodo, at accumsan ex ultrices. Integer eget ex eget dui cursus euismod et accumsan felis. Nullam laoreet sodales dui, ut finibus elit varius a. Sed elementum orci quis libero ullamcorper, eget egestas enim convallis. Sed nibh libero, tincidunt ultricies nibh quis, lobortis placerat mauris. Maecenas at laoreet risus, nec dictum libero. Donec accumsan, orci eu tempus mattis, nisl arcu auctor turpis, ac sollicitudin justo orci nec nulla.

Nam eget sem sed ligula vehicula iaculis. In non arcu a nisl interdum gravida. Nam egestas erat non turpis sagittis vestibulum. Praesent est metus, facilisis eu commodo sed, sagittis et est. Duis scelerisque luctus erat, elementum pulvinar felis bibendum a. Morbi hendrerit rhoncus consectetur. Vestibulum nec odio finibus, blandit turpis eget, dignissim orci. Curabitur eu ligula auctor, porttitor nulla non, maximus turpis. Nunc sed quam at est varius interdum eu vitae odio. Vestibulum egestas dapibus nulla sit amet fermentum.

Vestibulum ut neque urna. Ut nec odio lobortis, ultricies nulla quis, ultricies tellus. Nam ac iaculis sapien. Vivamus vitae risus id tortor interdum pellentesque. Quisque lorem lectus, sagittis vel metus et, sagittis finibus justo. Curabitur pulvinar odio tellus, eu vehicula est dictum eget. Morbi sed justo justo. Vivamus enim nibh, facilisis pretium luctus quis, ullamcorper quis ipsum. Pellentesque a mi a elit euismod malesuada.

Vestibulum interdum est vel orci tincidunt auctor. Nunc tristique nulla nec blandit fermentum. Maecenas id libero ut justo dictum sodales. Nullam justo sapien, dignissim vel enim at, porta pharetra metus. Integer euismod quam eget ligula gravida euismod. Pellentesque commodo, quam sit amet bibendum tempor,

nisi odio varius mauris, et accumsan justo ex sed nunc. Cras bibendum nibh ac dolor volutpat, non elementum orci pulvinar. Maecenas et porttitor nulla. Suspendisse sapien massa, dapibus at blandit et, rhoncus suscipit velit. Fusce molestie, velit eget sagittis suscipit, est libero aliquam libero, in iaculis mi tellus ac nunc.

1.2 Contribution

Sed in rhoncus lectus. Mauris vulputate purus non malesuada pulvinar. Curabitur ullamcorper hendrerit elit, id vulputate libero sagittis vel. Pellentesque ac faucibus est. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Integer venenatis, nisl eleifend pellentesque consequat, sem tortor malesuada ante, ut tincidunt elit tortor sit amet nunc.

Cras vehicula ipsum sit amet dui rutrum ultrices. Integer eu eleifend odio. Praesent tempor, libero id ullamcorper euismod, lectus diam lobortis mauris, id venenatis arcu sem vitae purus. Pellentesque luctus tristique metus quis mollis. Praesent ullamcorper neque velit, sed iaculis est convallis sit amet. Quisque nec massa ut magna lobortis imperdiet. Quisque rhoncus purus eget mollis aliquet. Donec vehicula viverra nisl, sed posuere turpis vulputate non. Donec malesuada, eros id interdum volutpat, ipsum orci luctus quam, non pulvinar urna ipsum eget purus. Nam hendrerit condimentum tristique.

Proin metus velit, tempor at fringilla non, dictum eu felis. Proin volutpat enim ut fermentum aliquam. Nam dictum nisi eu nisl viverra fermentum. Pellentesque tristique arcu non orci congue faucibus. Fusce sit amet nisl fringilla, feugiat turpis vitae, eleifend ante. Suspendisse elementum, lectus non pulvinar bibendum, lectus massa faucibus turpis, vitae porta risus sem quis metus. Maecenas id sapien et dui lobortis imperdiet nec eu mi. Quisque porttitor tincidunt nisi, eget sagittis orci. Nunc mattis erat malesuada facilisis viverra. Maecenas sodales iaculis nisi vel tincidunt. Morbi aliquet nibh ac facilisis consectetur. In ultrices libero quis massa porttitor cursus. Quisque suscipit ac tortor eget aliquet. Ut eget lacus vel orci viverra maximus at at purus.

Nam massa neque, varius nec suscipit id, cursus ac mi. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In hac habitasse platea dictumst. Vivamus facilisis nunc quis dictum consectetur. Sed congue sed magna non auctor. Vestibulum accumsan sit amet erat non congue. Sed at condimentum mi, sed scelerisque urna. Etiam tristique pulvinar rutrum. Donec semper nulla vitae rutrum semper. Maecenas ultrices nibh at orci sodales tincidunt sit amet vitae arcu. Curabitur interdum tincidunt ipsum, nec tincidunt nunc dapibus in. Nunc sit amet elementum massa, ut ornare lacus. Vivamus convallis fringilla erat, non suscipit sapien convallis eu. Nunc viverra lectus sit amet turpis viverra, eget iaculis purus rhoncus.

Morbi eu lectus arcu. Sed fringilla dui ut magna commodo, a malesuada ante pellentesque. Donec ornare facilisis pellentesque. Nulla vitae fringilla velit. Nunc id tellus nisl. Maecenas pretium elit lectus, nec consectetur nunc vulputate et. Sed facilisis magna nec gravida hendrerit. Sed a cursus nisl, in rhoncus massa. Curabitur ut nibh interdum, tempor risus vel, scelerisque nibh. Mauris quis ipsum sed risus tempor convallis ut a eros.

Alles Gescheite ist schon gedacht worden.
Man muss nur versuchen, es noch einmal zu denken.

All intelligent thoughts have already been thought;
what is necessary is only to try to think them again.

— Johann Wolfgang von Goethe
von_goethe_wilhelm_1829

2 Background

Contents

2.1 Introduction

This document introduction won't serve as a complete primer on LaTeX. There are plenty of those online, and googling your questions will often get you answers, especially from `http://tex.stackexchange.com`.

Instead, let's talk a little about a few of the features and packages lumped into this template situation. The `savequote` environment at the beginning of chapters can add some wittiness to your thesis. If you don't like the quotes, just remove that block.

For when it comes time to do corrections, there are two useful commands here. First, the `mccorrect` command allows you to highlight a short correction

like this one. When the thesis is typeset normally, the correction will just appear as part of the text. However, when you declare `\correctionstrue` in the main `Oxford_Thesis.tex` file, that correction will be highlighted in blue. That might be useful for submitting a post-viva, corrected copy to your examiners so they can quickly verify you've completed the task.

For larger chunks, like this paragraph or indeed entire figures, you can use the `mccorrection` environment. This environment highlights paragraph-sized and larger blocks with the same blue colour.

Read through the `Oxford_Thesis.tex` file to see the various options for one- and two-sided printing, including or excluding the separate abstract page, and turning corrections and draft footer on or off, and the separate option to centre your text on the page (for PDF submission) or offset it (for binding). There is also a separate option for master's degree submissions, which changes identifying information to candidate number and includes a word count. (Unfortunately, LaTeXhas a hard time doing word counts automatically, so you'll have to enter the count manually if you require this.)

2.2 Cardiac Imaging

Within months of Röntgen's discovery of the X-ray in 1895**gagliardi_rontgen_1996**, cardiac pathology was being investigated via non-invasive imaging **gagliardi_cardiac_1996**. Over the intervening years, cardiac imaging modalities and techniques have advanced significantly. Clinically, cardiac imaging is used for two broad purposes: diagnosis of pathophysiology and guidance of interventional procedures. These applications impose different requirements on imaging equipment, image acquisition time, computational complexity, spatial and temporal resolution, and tissue discrimination. The common diagnostic and interventional cardiac imaging techniques in current clinical practice are reviewed below. An accessible introduction to the physics of medical imaging can be found in Webb's *Introduction to Biomedical Imaging* **webb_introduction_2002**. A comprehensive overview of the use of

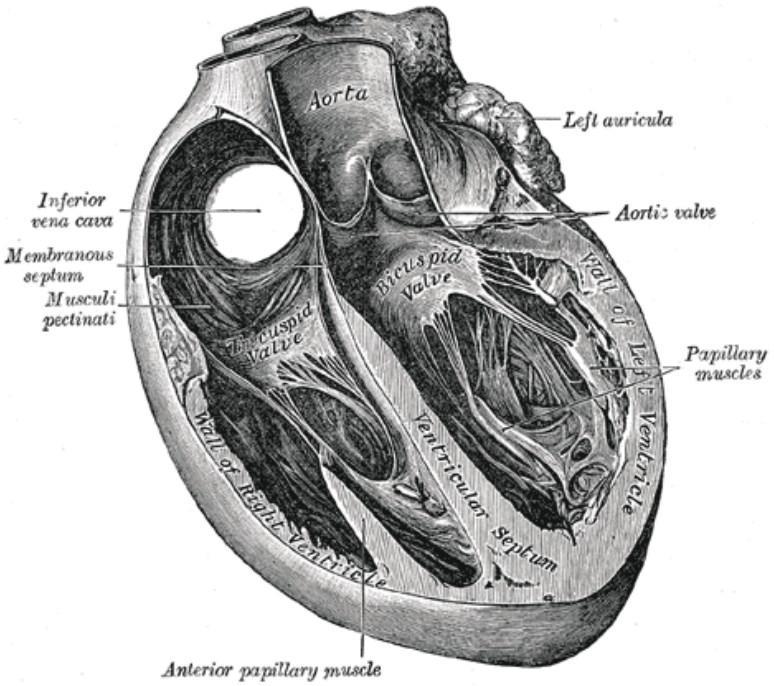


Figure 2.1: Four-chamber illustration of the human heart. Clockwise from upper-left: right atrium, left atrium, left ventricle, right ventricle.

imaging in clinical cardiology is presented in Leeson's *Cardiovascular Imaging* **leeson_cardiovascular_2011**.

2.2.1 Diagnostic Imaging

Beyond the chest X-ray ('plain film'), the key non-invasive imaging modalities in diagnostic cardiology are echocardiography, magnetic resonance imaging, and X-ray computed tomography, which are reviewed below. Nuclear medicine, including positron emission tomography (PET) and single-photon emission computed tomography (SPECT), are not discussed here, as they do not play a role in the chapters to follow.

Echocardiography

The use of acoustic waves for medical diagnosis, inspired by naval sonar, was initially developed in the 1940s **gagliardi_ultrasonography_1996**. By 1954, the first clinically useful cardiac ultrasound – examining motion of the mitral valve in stenosis – was reported **edler_ultrasonic_1957**. These early scans were one-dimensional im-

ages ('A-mode'), sometimes repeated to generate a time axis ('M-mode'). The sector-scanning probe was developed in the 1970s **bom_ultrasonic_1971**; **griffith_sector_1974**, leading to the 'B-mode' that a modern cardiologist would recognise as an echocardiogram.

3

Hierarchical Attentive Recurrent Tracking

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 auxiliary tasks relevant for tracking. Evaluation of the proposed model is performed on two datasets: pedestrian tracking on the KTH activity recognition dataset and the more difficult KITTI object tracking dataset.

3.1 Introduction

In computer vision, designing an algorithm for model-free tracking of anonymous objects is challenging, since no target-specific information can be gathered *a priori* and yet the algorithm 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, which exposes them to noise and increases the associated computational cost or they 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*.

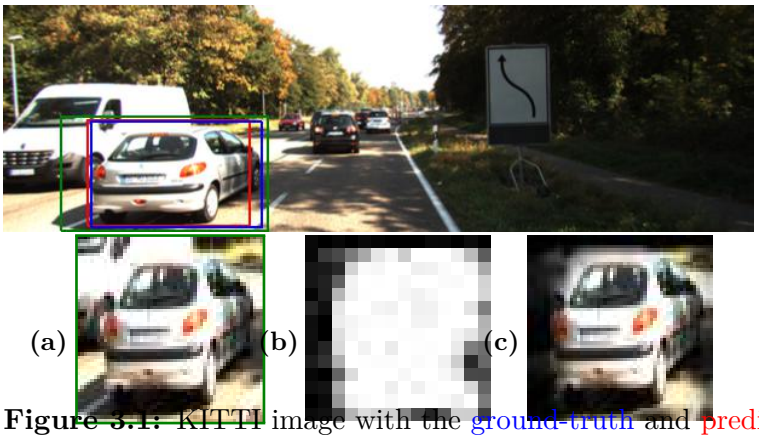


Figure 3.1: KITTI image with the ground-truth and predicted bounding boxes and an attention glimpse. The lower row corresponds to the hierarchical attention of our model: 1^{st} layer extracts an attention glimpse (a), the 2^{nd} layer uses appearance attention to build a location map (b). The 3^{rd} layer uses the location map to suppress distractors, visualised in (c).

Attention mechanisms have recently been explored in machine learning in a wide variety of contexts **Vinyals2014**; **Jaderberg2015**, often providing new capabilities to machine learning algorithms **Graves2016**; **Wierstra2015draw**; **Eslami2016**. While they improve efficiency **Graves2014recurrent** and performance on state-of-the-art machine learning benchmarks **Vinyals2014**, their architecture is much

simpler than that of the mechanisms found in the human visual cortex **Dayan2001**. Attention has also been long studied by neuroscientists **Ungerleider2000**, who believe that it is crucial for visual perception and cognition **Olshausen2016foveal**, 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 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 with respect to 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 primary visual 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 1.3). 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 1.1. The first stage immediately discards spatially irrelevant input, while later stages focus on producing target-specific filters to emphasise visual features *particular* to the object of interest.

The resulting framework is end-to-end trainable and we resort to maximum likelihood estimation (MLE) for parameter learning. 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 $\mathbf{x}_{1:T} \in \mathbb{R}^{H \times W \times C}$, where the superscript denotes height, width and the number of channels of the image, respectively, 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

$$p(\mathbf{b}_{2:T} \mid \mathbf{x}_{1:T}, \mathbf{b}_1) = \int p(\boldsymbol{h}_1 \mid \mathbf{x}_1, \mathbf{b}_1) \prod_{t=2}^{T} \int p(\mathbf{b}_t \mid \boldsymbol{h}_t) p(\boldsymbol{h}_t \mid \mathbf{x}_t, \mathbf{b}_{t-1}, \boldsymbol{h}_{t-1}) \, \mathrm{d}\boldsymbol{h}_t \, \mathrm{d}\boldsymbol{h}_1, \tag{3.1}$$

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 (Section 6.2) before describing how information flows through the components of our hierarchical attention in Section 1.3. Section 1.4 details the losses applied to guide the attention. Section 1.5 presents experiments on KTH and KITTI datasets with comparison to related attention-based trackers. Section 6.5 discusses the results and intriguing properties of our framework and Section 1.7 concludes the work. Code and results are available online[1].

3.2 Related Work

A number of recent studies have demonstrated that visual content can be captured through a sequence of spatial glimpses or foveation **Graves2014recurrent**;

[1] `https://github.com/akosiorek/hart`

Wierstra2015draw. 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, **Olshausen2016foveal** show that if spatial attention is capable of zooming, a regular grid sampling is sufficient. **Jaderberg2015** introduced the spatial transformer network (STN) which provides a fully differentiable means of transforming feature maps, conditioned on the input itself. **Eslami2016** 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, **Eslami2016** 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 **Kahou2015ratm** 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 **Stollenga2014**. **Brabandere2016dfn** 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. **Karl2017** 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 **Kahou2015ratm**. **Cheung2016gtc** applied STNs **Jaderberg2015** as attention mechanisms for real-world object tracking, but failed due to exploding gradients potentially arising from the difficulty of the data. **Ning2016** 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 **Held2016**; **Valmadre2017**. Both methods explicitly process small search areas around the previous target position to produce a bounding box offset **Held2016** or a correlation response map with the maximum corresponding to the target position **Valmadre2017**. We acknowledge the recent work[2] of **Gordon2017** 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 **Kahou2015ratm** 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.

3.3 Hierarchical Attention

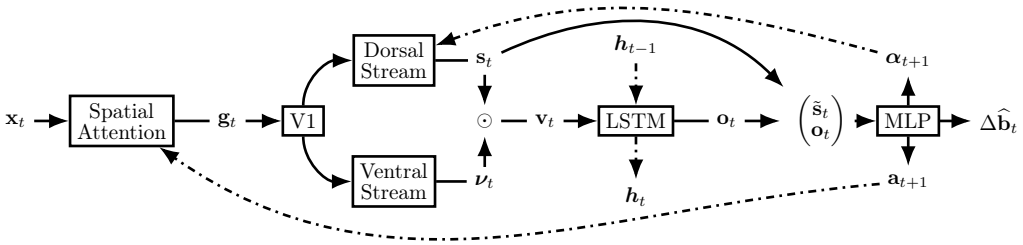


Figure 3.2: Hierarchical Attentive Recurrent Tracking. Spatial attention extracts a glimpse $\mathbf{g}_t$ from the input image $\mathbf{x}_t$. V1 and the ventral stream extract appearance-based features $\boldsymbol{\nu}_t$ while the dorsal stream computes a foreground/background segmentation $\mathbf{s}_t$ of the attention glimpse. 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.

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 were known, it would be possible to compute the expected location of the object at the next time step.

[2] **Gordon2017** only became available at the time of submitting this paper.

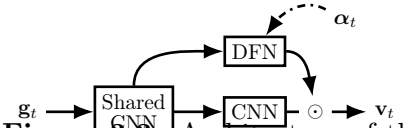


Figure 3.3: Architecture of the appearance attention. V1 is implemented as a CNN shared among the dorsal stream (DFN) and the ventral stream (CNN). The $\odot$ symbol represents the Hadamard product and implements masking of visual features by the foreground/background segmentation.

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 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 **Ungerleider2000**. We now describe the individual components of the system.

Spatial Attention Our spatial attention mechanism is similar to the one used by **Kahou2015ratm**. 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

$$\mathbf{g}_t = \mathbf{A}_t^y \mathbf{x}_t \left(\mathbf{A}_t^x\right)^\mathsf{T}. \tag{3.2}$$

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 **Kahou2015ratm**, 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 (compared to 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 **Dayan2001**. 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 $\boldsymbol{\psi}_t^{a \times b \times c \times d}$, where the superscript denotes the size of the filters and the number of input and output feature maps, as

$$\boldsymbol{\Psi}_t = \left\{ \boldsymbol{\psi}_t^{a_i \times b_i \times c_i \times d_i} \right\}_{i=1}^{K} = \mathrm{MLP}\, \boldsymbol{\alpha}_t. \tag{3.3}$$

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]

$$\mathbf{v}_t = \text{MLP}\,\text{vec}(\boldsymbol{\nu}_t \odot \mathbf{s}_t), \tag{3.4}$$

with $\odot$ being the Hadamard product.

State Estimation Our approach relies on being able to predict future object appearance and location, and therefore it heavily depends on 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.

$$\mathbf{o}_t, \boldsymbol{h}_t = \text{LSTM}(\mathbf{v}_t, \boldsymbol{h}_{t-1}), \tag{3.5}$$

$$\boldsymbol{\alpha}_{t+1}, \Delta\mathbf{a}_{t+1}, \Delta\widehat{\mathbf{b}}_t = \text{MLP}\,\mathbf{o}_t, \text{vec}(\mathbf{s}_t), \tag{3.6}$$

$$\mathbf{a}_{t+1} = \mathbf{a}_t + \tanh(\boldsymbol{c})\Delta\mathbf{a}_{t+1}, \tag{3.7}$$

$$\widehat{\mathbf{b}}_t = \mathbf{a}_t + \Delta\widehat{\mathbf{b}}_t \tag{3.8}$$

Equations (1.5) to (1.8) 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 1.4), the bounding box $\widehat{\mathbf{b}}_t$ is estimated relative to attention at time t.

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

3.4 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 **Jaderberg2016**, 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

$$\mathcal{L}_{\text{HART}}(\mathcal{D}, \theta) = \lambda_{\text{t}}\mathcal{L}_{\text{t}}(\mathcal{D}, \theta) + \lambda_{\text{s}}\mathcal{L}_{\text{s}}(\mathcal{D}, \theta) + \lambda_{\text{a}}\mathcal{L}_{\text{a}}(\mathcal{D}, \theta) + R(\boldsymbol{\lambda}) + \beta R(\mathcal{D}, \theta), \quad (3.9)$$

with dataset $\mathcal{D} = \{(\mathbf{x}_{1:T}, \mathbf{b}_{1:T})^i\}_{i=1}^M$, network parameters θ, regularisation terms $R(\cdot)$, adaptive weights $\boldsymbol{\lambda} = \{\lambda_{\text{t}}, \lambda_{\text{s}}, \lambda_{\text{d}}\}$ 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 $\text{IoU}(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a} \cap \boldsymbol{b}}{\boldsymbol{a} \cup \boldsymbol{b}} = \frac{\text{area of overlap}}{\text{area of union}}$. 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 distribution (as it cannot be normalised), it is often used for evaluation **VOT2016**. We follow the work by **yu2016unitbox** and express the loss term as the negative log of IoU:

$$\mathcal{L}_{\text{t}}(\mathcal{D}, \theta) = \mathbb{E}_{p\left(\widehat{\mathbf{b}}_{1:T} | \mathbf{x}_{1:T}, \mathbf{b}_1\right)}\Big[-\log \text{IoU}(\widehat{\mathbf{b}}_t, \mathbf{b}_t)\Big], \quad (3.10)$$

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, where the logarithmic arguments are appropriately clipped to avoid numerical instabilities:

$$\mathcal{L}_{\mathrm{s}}(\mathcal{D},\theta) = \mathbb{E}_{p(\mathbf{a}_{1:T}|\mathbf{x}_{1:T},\mathbf{b}_1)}\left[-\log\left(\frac{\mathbf{a}_t \cap \mathbf{b}_t}{\mathrm{area}(\mathbf{b}_t)}\right) - \log(1-\mathrm{IoU}(\mathbf{a}_t,\mathbf{x}_t))\right]. \quad (3.11)$$

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

$$\mathcal{L}_{\mathrm{a}}(\mathcal{D},\theta) = \mathbb{E}_{p(\mathbf{a}_{1:T},\mathbf{s}_{1:T}|\mathbf{x}_{1:T},\mathbf{b}_1)}[H(\tau(\mathbf{a}_t,\mathbf{b}_t),\mathbf{s}_t)]. \quad (3.12)$$

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

Adaptive Loss Weights To avoid hyper-parameter tuning, we follow the work by **Kendall2017adaptive** 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: $R(\boldsymbol{\lambda}) = -\sum_i \log(\boldsymbol{\lambda}_i^{-1})$.

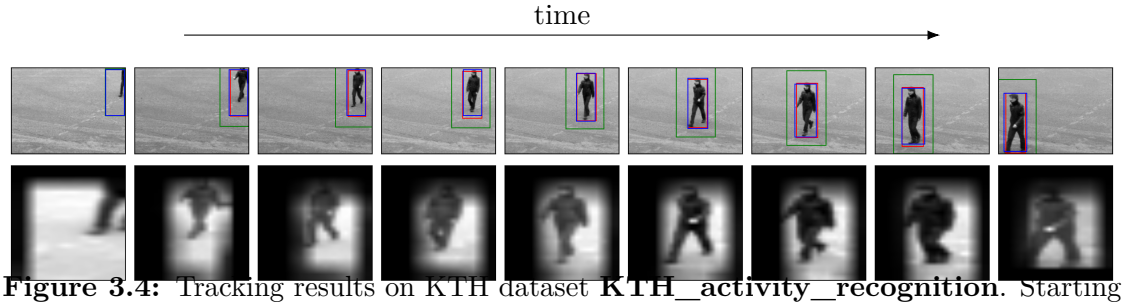


Figure 3.4: Tracking results on KTH dataset **KTH_activity_recognition**. Starting with the first initialisation frame where all three boxes overlap exactly, time flows from left to right showing every 16^{th} frame of the sequence captured at 25fps. The colour coding follows from Figure 1.1. The second row shows attention glimpses multiplied with appearance attention.

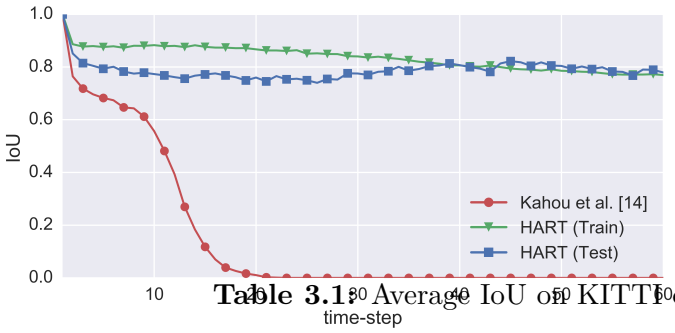
3.5 Experiments

3.5.1 KTH Pedestrian Tracking

Kahou2015ratm performed a pedestrian tracking experiment on the KTH activity recognition dataset **KTH_activity_recognition** 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 replicate the training procedure exactly, with the exception of using the RMSProp optimiser **Hinton2015RMSProp** 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 1.4 presents qualitative results.

3.5.2 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 **Geiger2013**. 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



Method	Avg. IoU
Kahou2015ratm	0.14
Spatial Att	0.60
App Att	0.78
HART	**0.81**

Table 3.1: Average IoU on KITTI over 60 time-steps.

images in this dataset are much more varied, we implement V1 as the first three convolutional layers of a modified AlexNet **Krizhevsky2012**. 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 **Krueger2016** with probability set to 0.05. The system was trained via curriculum learning **Bengio2009**, 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 1.1 and Figure 1.5 contain results of different variants of our model and of the RATM tracker by **Kahou2015ratm** 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

[4] `https://youtu.be/VvkjmOFRGSs`

(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 3.6: 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.

RATM tracker of **Kahou2015ratm** 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.

3.6 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 **Kahou2015ratm**, 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 1.6 shows examples of glimpses and corresponding location maps for a model with and without loss on the appearance attention. In Fig. 1.6a the model with loss on appearance attention is able to track a pedestrian even after it was occluded by another human. Figure 1.6b shows that, when not penalised, location map might not be very object-specific and can miss the object entirely (right-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 our 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.

3.7 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 offers the ability to attend each object in turn.

Acknowledgements

We would like to thank Oiwi Parker Jones and Martin Engelcke for discussions and valuable insights and Neil Dhir for his help with editing the paper. Additionally, we would like to acknowledge the support of the UK's Engineering and Physical Sciences Research Council (EPSRC) through the Programme Grant EP/M019918/1 and the Doctoral Training Award (DTA). The donation from Nvidia of the Titan Xp GPU used in this work is also gratefully acknowledged.

4

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

Abstract

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

4.1 Introduction

Autonomous vehicles need to operate in rich environments that contain a large variety of interacting object. This variety motivates the need for *class-agnostic* object trackers, which break with the popular tracking-by-detection paradigm **Zhang2008**; **Milan2014**; **bae2017confidence**; **keuper2018motion**. In tracking-by-detection, static video frames are first analysed by an object detector, e. g., a pre-trained deep convolutional neural network (CNN) such as YOLO (**Redmon15**), and then the detected objects are linked across frames. Algorithms from this family can achieve high accuracy, provided sufficient labelled data to train the object detector, and given that all encountered objects can be associated with known classes.

HART is a recently proposed alternative for single-object tracking (SOT), where an arbitrary object can be tracked from an initial video frame (**Kosiorek17**). Since the initial bounding-box is user-provided and may be placed over any part of the image, regardless of whether it corresponds to an object and its class, HART can track arbitrary objects. HART efficiently processes just the relevant part of an image using spatial attention; it also integrates object detection, feature extraction, and motion modelling into one network, which is trained fully end-to-end. Contrary to tracking-by-detection, where only one video frame is typically processed at any given time to generate bounding box proposals, end-to-end learning in HART allows discovering complex visual and spatio-temporal patterns in videos, which is conducive to inferring what an object is and how it moves.

In the original formulation, HART is limited to the single-object modality—as are other existing end-to-end trackers **Kahou15**; **Danesh19**; **Gordon2018**.

(a) HART **(b)** MOHART

Figure 4.1: Single-object tracking with HART (left) and our extension to multi-object tracking (MOHART, right). In our proposed framework, the different HART trackers are connected via a relational reasoning module allowing for more robust tracking and more accurate future trajectory prediction.

In this work, we present MOHART, a class-agnostic tracker with complex relational reasoning capabilities provided by a multi-headed self-attention module (**Vaswani17**; **Lee2019settransformer**). MOHART infers the latent state of every tracked object in parallel, and uses self-attention to inform per-object states about other tracked objects. This helps to avoid performance loss under self-occlusions of tracked objects or strong ego-motion. Moreover, since the model is trained end-to-end, it is able to learn how to manage faulty or missing sensor inputs. It can also use the inferred objects' states to predict their future trajectories, which depend on interactions between different objects. See Fig. 2.1 for a high-level illustration of HART and MOHART.

After describing related work in Section 2.2 and the methodology in Section 6.3, we employ the algorithm on toy domains to validate its efficacy in Section 2.4. By controlling the stochasticity of toy environments, we show that single-object tracking is sufficient in some cases, even those featuring strong long-range interactions, while it may fail in other cases. This may hint at a similar phenomenon in the real world: tracking objects or predicting their future motion independently may be possible in most (but not all) cases, while solving the remaining corner cases might require taking interactions between objects into account. It is these corner cases that motivate our work. In Section 2.5, we test MOHART on three real world datasets (MOTChallenge **MOT16**, UA-DETRAC **Wen15**, Stanford Drone dataset **DroneDataset**) and show that relational reasoning between objects is most important on the MOTChallenge dataset. We hypothesise that this is due to its richness in ego-motion, occlusions and crowded scenes—a result supported by our ablation study. Furthermore, we show that MOHART is able to gracefully handle missing sensory inputs—without any architectural changes. In this case, it falls back on its internal motion model, which also allows for accurate prediction of object locations multiple time steps into the future, learned in a data-driven manner.

4.2 Related Work

Tracking-by-Detection Vision-based tracking approaches typically follow a tracking-by-detection paradigm: objects are first detected in each frame independently, and then a tracking algorithm links the detections from different frames to propose a coherent trajectory **Zhang2008**; **Milan2014**; **bae2017confidence**; **keuper2018motion**. Motion models and appearance are often used to improve the association between detected bounding-boxes and multiple trackers in a postprocessing step. Recently, elements of this pipeline have been replaced with learning-based approaches such as deep learning **Nam2016**; **Ning2017**; **keuper2018motion**; **bae2017confidence** or reinforcement learning **Xiang2015**. Some approaches are targeted towards robustness across domains, for example by using a category-agnostic object detector and performing classification only in a post-processing step **Osep2017**; **Posner2016**.

End-to-End Tracking A newly established and much less explored stream of work approaches tracking in an end-to-end fashion. A key difficulty here is that extracting an image crop (according to bounding-boxes provided by a detector), is non-differentiable and results in high-variance gradient estimators. **Kahou15** propose an end-to-end tracker with soft spatial-attention using a 2D grid of Gaussians instead of a hard bounding-box. HART draws inspiration from this idea, employs an additional attention mechanism, and shows promising performance on the real-world KITTI dataset **Kosiorek17**. HART, which forms the foundation of this work, is explained in detail in Section 6.3. It has also been extended to incorporate depth information from RGBD cameras **Danesh19**. **Gordon2018** propose an approach in which the crop corresponds to the scaled up previous bounding-box. This simplifies the approach, but does not allow the model to learn where to look— i. e., no gradient is backpropagated through crop coordinates. To the best of our knowledge, there are no successful implementations of any such end-to-end approaches for multi-object tracking beyond SQAIR (**Kosiorek2018sqair**), which works only on datasets with static backgrounds. On real-world data, the only end-to-end approaches correspond

to applying multiple single-object trackers in parallel—a method which does not leverage the potential of scene context or inter-object interactions.

Pedestrian trajectory prediction Predicting pedestrian trajectories has a long history in computer vision and robotics. Initial research modelled social forces using hand-crafted features **UCY**; **ETH**; **IGP**; **yamaguchi2011you** or MDP-based motion transition models **rudenko2018joint**, while more recent approaches learn from context information, e. g., positions of other pedestrians or landmarks in the environment. Social-LSTM **social-lstm** employs a long short-term memory (LSTM) to predict pedestrian trajectories and uses max-pooling to model global social context. Attention mechanisms have been employed to query the most relevant information, such as neighbouring pedestrians, in a learnable fashion **su2016crowd**; **fernando2018soft**; **sadeghian2019sophie**. Apart from relational learning, context **varshneya2017human**, periodical time information **sun20183dof**, and constant motion priors **scholler2019simpler** have proven effective in predicting long-term trajectories.

Our work stands apart from this prior art by not relying on ground truth tracklets. Instead, it addresses the more challenging task of working directly with visual input, performing tracking, modelling interactions, and, depending on the application scenario, simultaneously predicting future motions. As such, it can also be compared to Visual Interaction Networks (VIN) **Watters2017**, which use a CNN to encode three consecutive frames into state vectors—one per object—and feeds these into a RNN, which has an Interaction Network **Battaglia2016** at its core. More recently, Relational Neural Expectation Maximization (R-NEM) has been proposed as an unsupervised approach which combines scene segmentation and relational reasoning **Steenkiste2018**. Both VINs and R-NEM are able to make accurate predictions in physical scenarios, but, to the best of our knowledge, have not been applied to real world data.

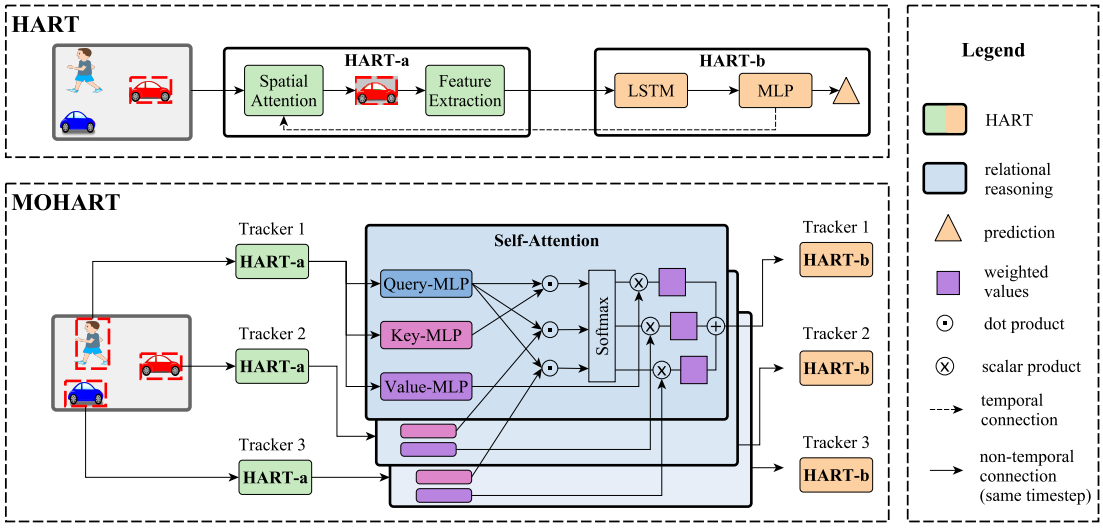


Figure 4.2: Single-object tracking with HART (top) and our extension to multi-object tracking (MOHART, bottom). We track multiple objects in a scene by applying multiple HART trackers in parallel. The trackers can exchange information about their respective objects using query-key-value attention. To do so, HART is split into HART-a and HART-b (see top row) and we apply the self-attention mechanism in-between the parts.

4.3 Recurrent Multi-Object Tracking with Self-Attention

We start by describing the hierarchical attentive recurrent tracking (HART) algorithm **Kosiorek17**, and then follow with an extension of HART to tracking multiple objects, where multiple instances of HART communicate with each other using multi-headed attention to facilitate relational reasoning. We also explain how this method can be extended to trajectory prediction instead of just tracking.

Hierarchical Attentive Recurrent Tracking (hart) HART is an attention-based recurrent algorithm, which can efficiently track single objects in a video. It uses a spatial attention mechanism to extract a *glimpse* $\mathbf{g}_t$, which corresponds to a small crop of the image $\mathbf{x}_t$ at time-step t, containing the object of interest. This allows it to dispense with the processing of the whole image and can significantly decrease the amount of computation required. HART uses a CNN to convert the glimpse $\mathbf{g}_t$ into features $\mathbf{f}_t$, which then update the hidden state $\mathbf{h}_t$ of a LSTM core. The hidden state is used to estimate the current bounding-box $\mathbf{b}_t$, spatial attention parameters for the next time-step $\mathbf{a}_{t+1}$, as well as object appearance. Importantly, the recurrent

core can learn to predict complicated motion conditioned on the past history of the tracked object, which leads to relatively small attention glimpses—contrary to CNN-based approaches (**Held2016goturn**; **Valmadre2017corr**), HART does not need to analyse large regions-of-interest to search for tracked objects. In the original paper, HART processes the glimpse with an additional ventral and dorsal stream on top of the feature extractor. Early experiments have shown that this does not improve performance on the MOTChallenge dataset, presumably due to the oftentimes small objects and overall small amount of training data. Figure 2.2 illustrates HART, further details are provided in Section 2.A.

The algorithm is initialised with a bounding-box[1] $\mathbf{b}_1$ for the first time-step, and operates on a sequence of raw images $\mathbf{x}_{1:T}$. For time-steps $t \geq 2$, it recursively outputs bounding-box estimates for the current time-step and predicted attention parameters for the next time-step. The performance of both algorithms is measured as intersection-over-union (IoU) averaged over all time steps in which an object is present, excluding the first time step.

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

Multi-Object Hierarchical Attentive Recurrent Tracking (mohart) Multi-object support in HART requires the following modifications. Firstly, in order to handle a dynamically changing number of objects, we apply HART to multiple objects in parallel, where all parameters between HART instances are shared. We refer to each HART instance as a *tracker*. Secondly, we introduce a presence variable $p_{t,m}$ for object m. It is used to mark whether an object should interact with other objects, as well as to mask the loss function (described in **Kosiorek17**) for the given object when it is not present. In this setup, parallel trackers cannot exchange

[1] We can use either a ground-truth bounding-box or one provided by an external detector; the only requirement is that it contains the object of interest.

information and are conceptually still single-object trackers, which we use as a baseline, referred to as HART (despite it being an extension of the original algorithm). Finally, to facilitate communication between trackers, we augment HART with an additional step between feature extraction and the LSTM.

Let $\mathbf{f}_{t,m}$ be the feature vector extracted from the glimpse corresponding to the m^{th} object, and let $\mathbf{f}_{t,1:M}$ be the set of such features extracted from all glimpses. Since different objects can interact with each other, it is necessary to use a method that can inform each object about the effects of their interactions with other objects. Moreover, since features extracted from different objects comprise a set, this method should be permutation-equivariant, i. e., the results should not depend on the order in which object features are processed. Therefore, we use the multi-head self-attention block (SAB, **Lee2019settransformer**), which is able to account for higher-order interactions between set elements when computing their representations, thereby allowing rich information exchange, and it can do so in a permutation-equivariant manner. Intuitively, in our case, SAB allows any of the trackers to query other trackers about attributes of their respective objects, e. g., distance between objects, their direction of movement, or their relation to the robot. This is implemented as follows,

$$Q = W_q \mathbf{f}_{1:M} + b_q \,, \qquad K = W_k \mathbf{f}_{1:M} + b_k \,, \qquad V = W_v \mathbf{f}_{1:M} + b_v \,, \tag{4.1}$$

$$O_i = \mathrm{softmax}\left(Q_i K_i^T\right) V_i \,, \qquad i = 1, \ldots, H \,, \tag{4.2}$$

$$o_{1:M} = O = \mathrm{concat}(O_i, \ldots, O_H) \,, \tag{4.3}$$

where o_m is the output of the relational reasoning module for object m. Time-step subscripts are dropped to decrease clutter. In Equation (2.1), each of the extracted features $\mathbf{f}_{t,m}$ is linearly projected into a triplet of key $\mathbf{k}_{t,m}$, query $\mathbf{q}_{t,m}$ and value $\mathbf{v}_{t,m}$ vectors. Together, they comprise K, Q and V matrices with M rows and d_q, d_k, d_k columns, respectively. K, Q and V are then split up into multiple heads $H \in \mathbb{N}_+$, which allows to query different attributes by comparing and aggregating different projection of features. Multiplying $Q_i K_i^T$ in Equation (2.2) allows to compare every query vector $\mathbf{q}_{t,m,i}$ to all key vectors $\mathbf{k}_{t,1:M,i}$, where the value of the corresponding

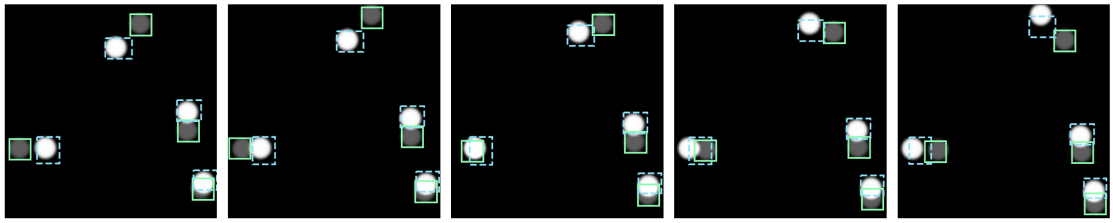


Figure 4.3: HART single object tracking applied four times in parallel. Dashed lines indicate spatial attention, solid lines are predicted bounding boxes at time step $T + 3$, faded circles show the ground truth location at $T + 3$. The repulsive force between each object pair scales with distance as $1/r$. There is no information exchange between the trackers and each tracker evidently only 'attends' to its own object. The fact that the future location is predicted accurately (i.e., much better than linear extrapolation) indicates that HART is able to capture complex motion patterns essentially allowing to draw conclusions about the force field. Shown are consecutive time steps from left to right.

dot-products represents the degree of similarity. Similarities are then normalised via a softmax operation and used to aggregate values V. Finally, outputs of different attention heads are concatenated in Equation (2.3). SAB produces M output vectors, one for each input, which are then concatenated with corresponding inputs and fed into separate LSTMs for further processing, as in HART—see Figure 2.2.

MOHART is trained fully end-to-end, contrary to other tracking approaches **Zhang2008**; **Milan2014**; **bae2017confidence**; **keuper2018motion**. It maintains a hidden state, which can contain information about the object's motion. One benefit is that in order to predict future trajectories, one can simply feed black frames into the model. Our experiments show that the model learns to fall back on the motion model captured by the LSTM in this case.

4.4 Validation on Simulated Data

To test the efficacy of the proposed algorithm, we conduct experiments on a toy domain. First, we show that HART as an end-to-end single-object tracker is able to capture complex motion patterns and leverage these to make accurate predictions. Second, we create a scenario which is not solvable for a single object tracker as it requires knowledge about the state of the other objects and relational reasoning. We show that MOHART, using self-attention for relational reasoning, is able to capture these interactions with high accuracy and compare it to other possible

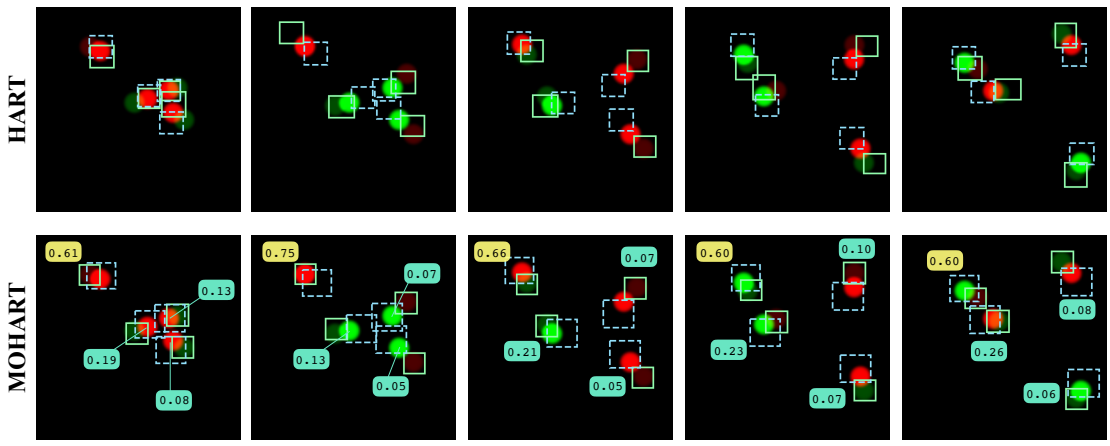


Figure 4.4: HART (top, 46% IoU) vs. MOHART (bottom, 76% IoU). Dashed lines show spatial attention, solid lines show predicted bounding boxes, faded circles indicate future ground truth locations. Circles of the same colour repel each other, circles of different colours attract each other. The colour coded identities are randomly assigned in each time step rendering information exchange between trackers (i.e. relational reasoning) necessary. The numbers in the bottom row indicate the self-attention weights from the perspective of the top left tracker (yellow number box).

implementations of MOHART (e.g., using max-pooling instead of self-attention). In order to accurately investigate the model's understanding of motion patterns and interactions between objects, in contrast to traditional tracking, the model is not trained to predict the current location of the object, but its location in a future time step. The domain we create for this purpose is a two dimensional squared box. It contains circular objects with approximated elastic collisions (energy and momentum conservation) between objects and with walls (see Figures 2.3 and 2.4).

In the first scenario (Figure 2.3), four circles each exert repulsive forces on each other, where the force scales with $1/r$, r being their distance. HART is applied four times in parallel and is trained to predict the location of each circle three time steps into the future. The different forces from different objects lead to a non-trivial force field at each time step. Predicting the future location just using the previous motion of one object (Figure 2.3 shows that each spatial attention box covers only the current object) accurately is therefore challenging. Surprisingly, the single object tracker solves this task with an average of 95% IoU over sequences of 15 time steps. This shows the efficacy of end-to-end tracking to capture complex motion patterns and use them to predict future locations. This, of course, could also be used to generate robust bounding boxes for a tracking task.

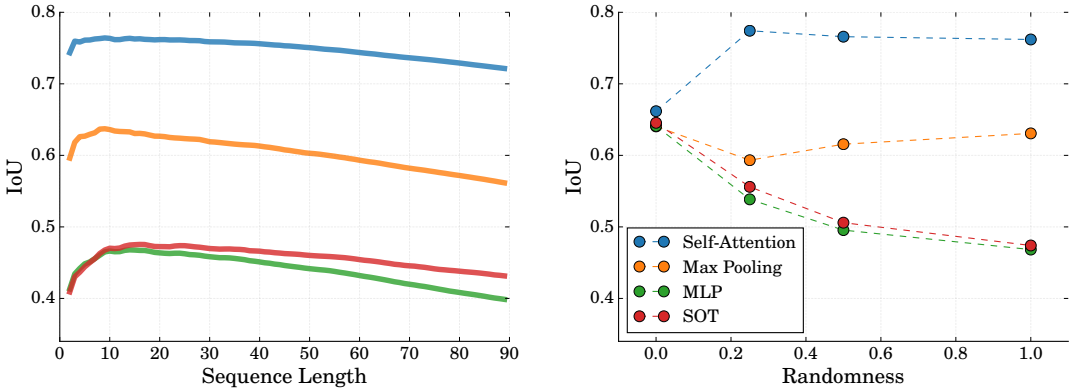


Figure 4.5: Left: average IoU over sequence length for different implementations of relational reasoning on the toy domain shown in Fig. 2.4 (randomness = 1.0). Right: performance depending on how often agents are re-assigned identities randomly (sequence length 15). The higher the randomness, the less static the force field is and the more vital relational reasoning is.

The second scenario (Figure 2.4) is constructed to be impossible to solve without exchanging information between objects. This is achieved by introducing two colour-coded identities. Agents of the same identity repel each other, agents of different identities attract each other. Crucially, each agent is randomly assigned its identity in each time step. Hence, the algorithm can no longer infer the forces exerted on one object without knowledge of the state of the other objects in the current time step. The forces in this scenario scale with $1/\sqrt{r}$ and the algorithm was trained to predict one time step into the future. HART is indeed unable to predict the future location of the objects accurately (Figure 2.4 - top). The achieved average IoU is 47%, which is only slightly higher than predicting the objects to have the same position in the next time step as in the current one (34%). A possible interpretation of the qualitative results (green boxes in Figure 2.4 - top) is that the model uses the momentum of each object to extrapolate into the future. This sometimes works well (bottom right object in frame 31) and sometimes not (top right object in frame 30). Using the relational reasoning module (Figure 2.4 - bottom), the model is now able to make meaningful predictions (76% IoU). Interestingly, in each frame, the attention scores have a strong correlation with the interaction strength (which directly scales with distance). Despite this not being necessary for the relational reasoning module, this is an interesting side-product as it did not receive any direct supervision.

Figure 2.5 (left) shows a quantitative comparison of augmenting HART with different relational reasoning modules when identities are re-assigned in every timestep (randomness = 1.0). Exchanging information between trackers of different objects in the latent space with an MLP leads to slightly worse performance than the SOT baseline, while simple max-pooling performs significantly better (ΔIoU $\sim 17\%$). This can be explained through the permutation invariance of the problem: the list of latent representation of the different objects has no meaningful order and the output of the model should therefore be invariant to the ordering of the objects. The MLP is in itself not permutation invariant and therefore prone to overfit to the (meaningless) order of the objects in the training data. Max-pooling, however, is permutation invariant and can in theory, despite its simplicity, be used to approximate any permutation invariant function - given a sufficiently large latent space **Zaheer2017**; **Wagstaff2019**. Max-pooling is often used to exchange information between different tracklets, e.g., in the trajectory prediction domain **social-lstm**; **Gupta2019**. However, self-attention, allowing for learned querying and encoding of information, solves the relational reasoning task significantly more accurately. In Figure 2.5 (right), the frequency with which object identities are reassigned randomly is varied. The results show that, in a deterministic environment, tracking does not necessarily profit from relational reasoning - even in the presence of long-range interactions. The less random, the more static the force field is and a static force field can be inferred from a small number of observations (see Figure 2.3). This does of course not mean that all stochastic environments profit from relational reasoning. What these experiments indicate is that tracking can not be expected to profit from relational reasoning by default in any environment, but instead in environments which feature (potentially non-deterministic) dynamics and predictable interactions.

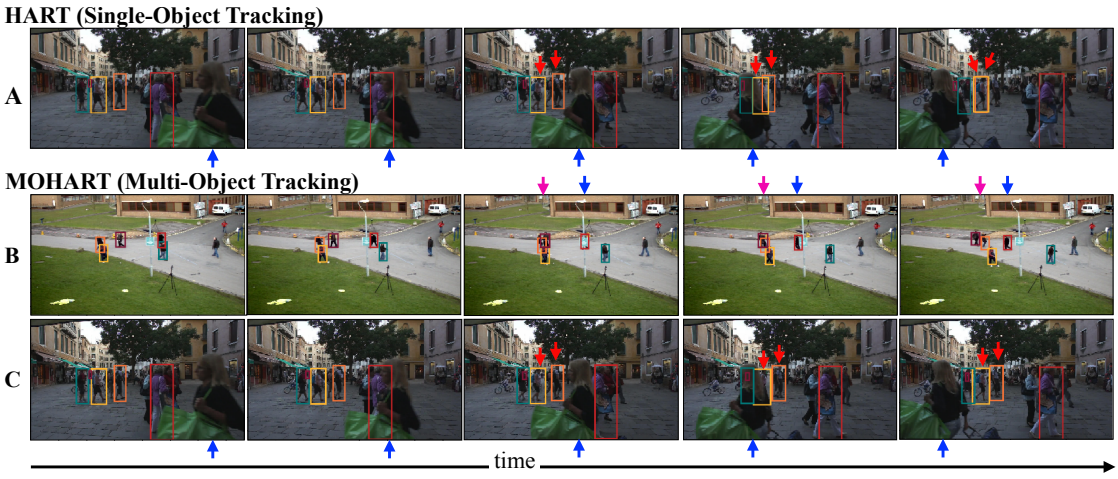


Figure 4.6: Tracking examples of both HART and MOHART. Coloured boxes are bounding boxes predicted by the model, arrows point at challenging aspects of the scenes. (A) & (C): Each person being tracked is temporarily occluded by a woman walking across the scene (blue arrows). MOHART, which includes a relational reasoning module, handles this more robustly (compare red arrows).

4.5 Relational Reasoning in Real-World Tracking

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

4.5.1 Experimental Details

We investigate three qualitatively different datasets: the MOTChallenge dataset **MOT16**, the UA-DETRAC dataset **Wen15**, and the Stanford Drone dataset **DroneDataset**. In order to increase scene dynamics and make the tracking/prediction problems more challenging, we sub-sample some of the high framerate scenes with a stride of two. Training and architecture details are given in Section 2.A and Section 2.B. We conduct experiments in three different modes:

Tracking. The model is initialised with the ground truth bounding boxes for a set of objects in the first frame. It then consecutively sees the following frames

and predicts the bounding boxes. The sequence length is 30 time steps and the performance is measured as intersection over union (IoU) averaged over the entire sequence excluding the first frame. This algorithm is either applied to the entire dataset or subsets of it to study the influence of certain properties of the data.

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

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

Table 4.1: Tracking performance on the MOTChallenge dataset measured in IoU.

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

Table 4.2: UA-DETRAC Dataset

	All	Crowded Scenes	Camera Black-out
MOHART	68.1%	**69.5%**	**64.2%**
HART	**68.4%**	68.6%	53.8%
Δ	-0.3%	0.9%	0.4%

Table 4.3: Stanford Drone Dataset

	All	Camera Black-out	CamBlack Bikes
MOHART	**57.3%**	**52.6%**	**53.3%**
HART	56.1%	53.3%	50.7%
Δ	1.2%	0.7%	2.6%

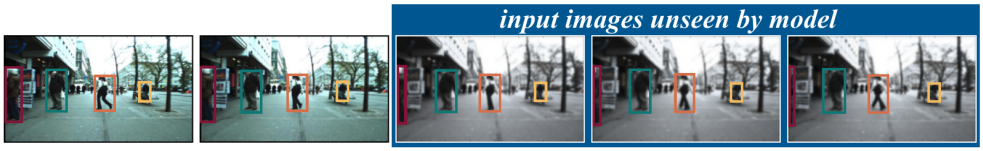
4.5.2 Results and Analysis

On the MOTChallenge dataset, HART achieves 66.6% intersection over union (see Table 2.1), which in itself is impressive given the small amount of training data of only 5225 training frames and no pre-training. MOHART achieves 68.5% (both

numbers are averaged over 5 runs, independent samples t-test resulted in $p < 0.0001$). The performance gain increases when only considering ego-motion data. This is readily explained: movements of objects in the image space due to ego-motion scenarios are correlated and can therefore be better understood when combining information from movements of multiple objects, i.e. performing relational reasoning. In another ablation, we filtered for only crowded scenes by requesting five objects to be present for, on average, 90% of the frames in a sub-sequence. For the MOT-Challenge dataset, this only leads to a minor increase of the performance gain of MOHART indicating that the dataset exhibits a sufficient density of objects to learn interactions. The biggest benefit from relational reasoning can be observed in the *camera blackout* experiments (setup explained in Section 2.5.1). Both HART and MOHART learn to rely on their internal motion models when confronted with black frames and propagate the bounding boxes according to the previous movement of the objects. It is unsurprising that this scenario profits particularly from relational reasoning. Qualitative tracking and *camera blackout* results are shown in Figure 2.6 and in Section 2.C, respectively.

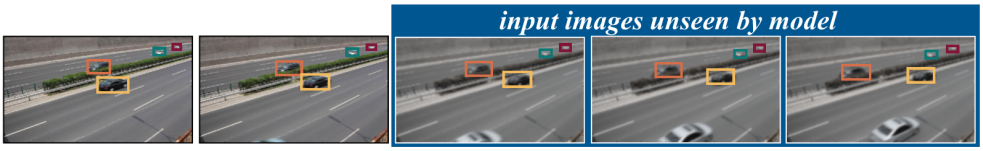
Tracking performance on the UA-DETRAC dataset only profits from relational reasoning when filtering for crowded scenes (see Table 2.2). The fact that the performance of MOHART is slightly worse on the vanilla dataset ($\Delta = -0.3\%$) can be explained with more overfitting. As there is no exchange between trackers for each object, each object constitutes an independent training sample.

The Stanford drone dataset (see Table 2.3) is qualitatively different to the other two as it is filmed from a top down view. The scenes are more crowded and each object only covers a small number of pixels rendering it a difficult problem for tracking. The dataset was designed for trajectory prediction, a problem setup where an algorithm is typically provided with ground truth tracklets in coordinate space and potentially an image as context information. The task is then to extrapolate these tracklets into the future. The tracking performance profits from relational reasoning more than on the UA-DETRAC dataset but less than on the



MOHART	–	**86.4%**	**80.0%**	**74.0%**	**68.4%**
HART	–	85.3%	79.2%	73.1%	67.4%
Momentum	–	–	78.2%	70.9%	63.9%

(a) Prediction results on the MOTChallenge dataset **MOT16**.



MOHART	–	**87.8%**	**81.3%**	**75.5%**	**70.0%**
HART	–	86.6%	79.6%	72.6%	66.1%
Momentum	–	-	80.5%	73.6%	67.2%

(b) Prediction results on the UA-DETRAC dataset (crowded scenes only) **Wen15**.

Figure 4.7: Peeking into the future. Only the first two frames are shown to the tracking algorithm followed by three black frames. MOHART learns to fall back on its internal motion model when no observation (i.e. only a black frame) is available. The reported IoU scores show the performance for the respective frames 0, 1, 2, and 3 time steps into the future.

MOTChallenge dataset. The performance gain on the *camera blackout* experiments are particularly strong when only considering cyclists.

In the results from the *prediction* experiments (see Figure 2.7) MOHART consistently outperforms HART. On both datasets, the model outperforms a baseline which uses momentum to linearly extrapolate the bounding boxes from the first two frames. This shows that even from just two frames, the model learns to capture motion models which are more complex than what could be observed from just the bounding boxes (i.e. momentum), suggesting that it uses visual information (HART & MOHART) as well as relational reasoning (MOHART). The strong performance gain of MOHART compared to HART on the UA-DETRAC dataset, despite the small differences for tracking on this dataset, can be explained as follows: this dataset features little interactions but strong correlations in motion. Hence when only having access to the first two frames, MOHART profits from estimating the velocities of multiple cars simultaneously.

4.6 Conclusion

With MOHART, we introduce an end-to-end multi-object tracker that is capable of capturing complex interactions and leveraging these for precise predictions as experiments both on toy and real world data show. However, the experiments also show that the benefit of relational reasoning strongly depends on the nature of the data. The toy experiments showed that in an entirely deterministic world relational reasoning was much less important than in a stochastic environment. Amongst the real-world dataset, the highest performance gains from relational reasoning were achieved on the MOTChallenge dataset, which features crowded scenes, ego-motion and occlusions.

Acknowledgements

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

Appendix

4.A Architecture Details

The architecture details were chosen to optimise HART performance on the MOTChallenge dataset. They deviate from the original HART implementation **Kosiorek17** as follows: The presence variable is predicted with a binary cross entropy loss. The maximum number of objects to be tracked simultaneously was set to 5 for the UA-DETRAC and MOTChallenge dataset. For the more crowded Stanford drone dataset, this number was set to 10. The feature extractor is a three layer convolutional network with a kernel size of 5, a stride of 2 in the first and last layer, 32 channels in the first two layers, 64 channels in the last layer, ELU activations, and skip connections. This converts the initial $32 \times 32 \times 3$ glimpse into a $7 \times 7 \times 64$ feature representation. This is followed by a fully connected layer with a 128 dimensional output and an elu activation. The spatial attention parameters are linearly projected onto 128 dimensions and added to this feature representation serving as a positional encoding. The LSTM has a hidden state size of 128. The self-attention unit in MOHART comprises linear projects the inputs to dimensionality 128 for each keys, queries and values. For the real-world experiments, in addition to the extracted features from the glimpse, the hidden states from the previous LSTM state are also fed as an input by concatinating them with the features. In all cases, the output of the attention module is concatenated to the input features of the respective object.

As an optimizer, we used RMSProp with momentum set to 0.9 and learning rate $5 * 10^{-6}$. For the MOTChallenge dataset and the UA-DETRAC dataset, the models were trained for 100,000 iterations of batch size 10 and the reported IoU is exponentially smoothed over iterations to achieve lower variance. For the Stanford

Figure 4..8: Camera blackout experiment on a pedestrian street scene from the MOTChallenge dataset without ego-motion. Subsequent frames are displayed going from top left to bottom right. Shown are the inputs to the model (some of them being black frames, i.e. arrays of zeroes) and bounding boxes predicted by MOHART (coloured boxes). This scene is particularly challenging as occlusion and missing sensor input coincide (fourth row).

Drone dataset, the batch size was increased to 32, reducing time to convergence and hence model training to 50,000 iterations.

4.B Experimental Details

The MOTChallenge and the UA-DETRAC dataset discussed in this section are intended to be used as a benchmark suite for multi-object-tracking in a tracking-by-detection paradigm. Therefore, ground truth bounding boxes are only available for the training datasets. The user is encouraged to upload their model which performs tracking in a data association paradigm leveraging the provided bounding box proposals from an external object detector. As we are interested in a different analysis (IoU given inital bounding boxes), we divide the training data further into training and test sequences. To make up for the smaller training data, we extend the MOTChallenge 2017 dataset with three sequences from the 2015 dataset (ETH-Sunnyday, PETS09-S2L1, ETH-Bahnhof). We use the first 70% of the frames of each of the ten sequences for training and the rest for testing. Sequences with

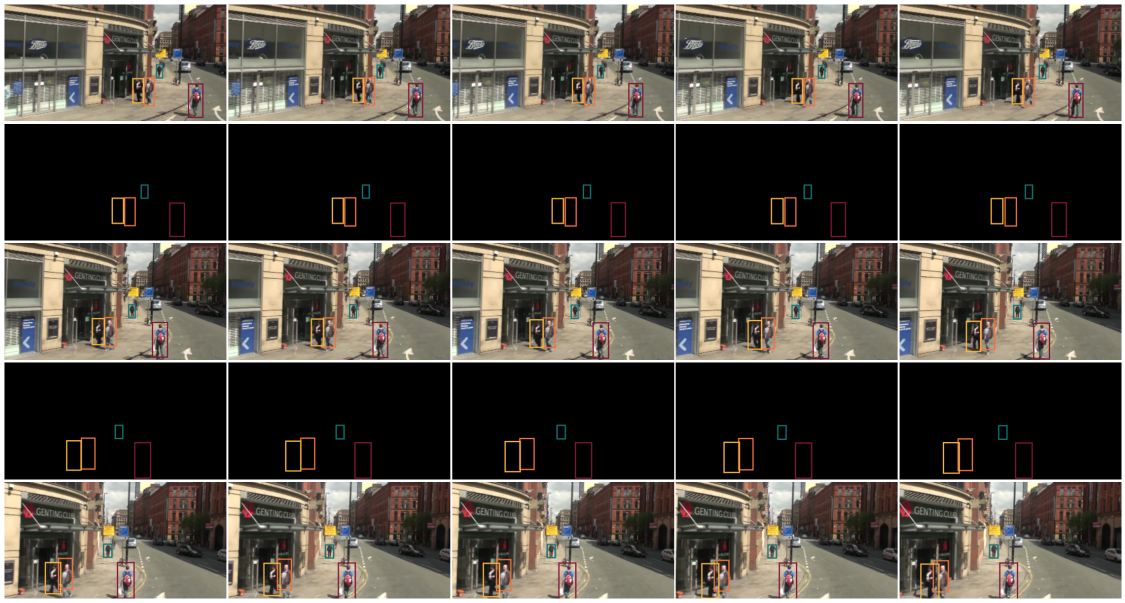


Figure 4.C.1: Camera blackout experiment on a street scene from the MOTChallenge dataset with strong ego-motion. The reader is encouraged to compare top left and bottom right frame to make the amount of ego-motion apparent.

high frame rates (30Hz) are sub-sampled with a stride of two. For the UA-DETRAC dataset, we split the 60 available sequences into 44 training sequences and 16 test sequences. For the considerably larger Stanford Drone dataset we took three videos of the scene *deathCircle* for training and the remaining two videos from the same scene for testing. The videos of the drone dataset were also sub-sampled with a stride of two to increase scene dynamics.

4.C Camera Blackout Experiments

In Section 2.5, we conducted a set of camera blackout experiments to test MOHART's capability of dealing with faulty sensor inputs. While traditional pipeline methods require careful consideration of different types of corner cases to properly handle erroneous sensor inputs, MOHART is able to capture these automatically, especially when confronted with similar issues in the training scenarios. To simulate this, we replace subsequences of the images with black frames. Figure 2..8 and Figure 2.C.1 show two such examples from the test data together with the model's prediction. MOHART learns not to update its internal model when confronted with black frames and instead uses the LSTM to propagate the bounding boxes. When proper sensor

input is available again, the model uses this to make a rapid adjustment to its predicted location and 'snap' back onto the object. This works remarkably well in both the presence of occlusion (Figure 2..8) and ego-motion (Figure 2.C.1). Tables 2.1 to 2.3 show that the benefit of relational reasoning is particularly high in these scenarios specifically. These experiments can also be seen as a proof of concept of MOHART's capabalities of predicting future trajectories—and how this profits from relational reasoning.

5

Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects

Abstract

We present Sequential Attend, Infer, Repeat (SQAIR), an interpretable deep generative model for videos of moving objects. It can reliably discover and track objects throughout the sequence of frames, and can also generate future frames conditioning on the current frame, thereby simulating expected motion of objects. This is achieved by explicitly encoding object presence, locations and appearances in the latent variables of the model. SQAIR retains all strengths of its predecessor, Attend, Infer, Repeat (AIR, **Eslami2016**), including learning in an unsupervised manner, and addresses its shortcomings. We use a moving multi-MNIST dataset to show limitations of AIR in detecting overlapping or partially occluded objects, and show how SQAIR overcomes them by leveraging temporal consistency of objects. Finally, we also apply SQAIR to real-world pedestrian CCTV data, where it learns to reliably detect, track and generate walking pedestrians with no supervision.

5.1 Introduction

The ability to identify objects in their environments and to understand relations between them is a cornerstone of human intelligence (**kemp2008discovery**). Arguably, in doing so we rely on a notion of spatial and temporal consistency which gives rise to an expectation that objects do not appear out of thin air, nor do they spontaneously vanish, and that they can be described by properties such as location, appearance and some dynamic behaviour that explains their evolution over time. We argue that this notion of consistency can be seen as an *inductive bias* that improves the efficiency of our learning. Equally, we posit that introducing such a bias towards spatio-temporal consistency into our models should greatly reduce the amount of supervision required for learning.

One way of achieving such inductive biases is through model structure. While recent successes in deep learning demonstrate that progress is possible without explicitly imbuing models with interpretable structure (**lecun2015deep**), recent works show that introducing such structure into deep models can indeed lead to favourable inductive biases improving performance e.g. in convolutional networks (**lecun1989backpropagation**) or in tasks requiring relational reasoning (**Santoro2017**). Structure can also make neural networks useful in new contexts by significantly improving generalization, data efficiency (**jacobsen2016struc**) or extending their capabilities to unstructured inputs (**Graves2016**).

AIR, introduced by **Eslami2016**, is a notable example of such a structured probabilistic model that relies on deep learning and admits efficient amortized inference. Trained without any supervision, AIR is able to decompose a visual scene into its constituent components and to generate a (learned) number of latent variables that explicitly encode the location and appearance of each object. While this approach is inspiring, its focus on modelling individual (and thereby inherently static) scenes leads to a number of limitations. For example, it often merges two objects that are close together into one since no temporal context is available to distinguish between them. Similarly, we demonstrate that AIR struggles to identify

partially occluded objects, e.g. when they extend beyond the boundaries of the scene frame (see Figure 3.4.4 in Section 3.4.1).

Our contribution is to mitigate the shortcomings of AIR by introducing a sequential version that models sequences of frames, enabling it to discover and track objects over time as well as to generate convincing extrapolations of frames into the future. We achieve this by leveraging temporal information to learn a richer, more capable generative model. Specifically, we extend AIR into a spatio-temporal state-space model and train it on unlabelled image sequences of dynamic objects. We show that the resulting model, which we name Sequential AIR (SQAIR), retains the strengths of the original AIR formulation while outperforming it on moving MNIST digits.

The rest of this work is organised as follows. In Section 3.2, we describe the generative model and inference of AIR. In Section 3.3, we discuss its limitations and how it can be improved, thereby introducing SQAIR, our extension of AIR to image sequences. In Section 6.4, we demonstrate the model on a dataset of multiple moving MNIST digits (Section 3.4.1) and compare it against AIR trained on each frame and VRNN of **Chung2015** with convolutional architectures, and show the superior performance of SQAIR in terms of log marginal likelihood and interpretability of latent variables. We also investigate the utility of inferred latent variables of SQAIR in downstream tasks. In Section 3.4.2 we apply SQAIR on real-world pedestrian CCTV data, where SQAIR learns to reliably detect, track and generate walking pedestrians without any supervision. Code for the implementation on the MNIST dataset[1] and the results video[2] are available online.

5.2 Attend, Infer, Repeat (AIR)

AIR, introduced by **Eslami2016**, is a structured VAE capable of decomposing a static scene $\mathbf{x}$ into its constituent objects, where each object is represented as a separate triplet of continuous latent variables $\mathbf{z} = \{\mathbf{z}^{\text{what},i}, \mathbf{z}^{\text{where},i}, z^{\text{pres},i}\}_{i=1}^{n}$, $n \in \mathbb{N}$

[1] code: github.com/akosiorek/sqair
[2] video: youtu.be/-IUNQgSLE0c

being the (random) number of objects in the scene. Each triplet of latent variables explicitly encodes position, appearance and presence of the respective object, and the model is able to infer the number of objects present in the scene. Hence it is able to count, locate and describe objects in the scene, all learnt in an unsupervised manner, made possible by the inductive bias introduced by the model structure.

Generative Model The generative model of AIR is defined as follows

$$p_\theta(n) = \mathrm{Geom}(n \mid \theta), \qquad p_\theta(\mathbf{z}^{\mathrm{w}} \mid n) = \prod_{i=1}^{n} p_\theta\Big(\mathbf{z}^{w,i}\Big) = \prod_{i=1}^{n} \mathcal{N}\Big(\mathbf{z}^{w,i} | \mathbf{0}, \mathbf{I}\Big),$$
$$p_\theta(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}\Big(\mathbf{x} \mid \mathbf{y}_t, \sigma_x^2 \boldsymbol{I}\Big), \qquad \text{with} \quad \mathbf{y}_t = \sum_{i=1}^{n} \mathrm{h}_\theta^{\mathrm{dec}}(\mathbf{z}^{\mathrm{what},i}, \mathbf{z}^{\mathrm{where},i}), \tag{5.1}$$

where $\mathbf{z}^{\mathrm{w},i} := (\mathbf{z}^{\mathrm{what},i}, \mathbf{z}^{\mathrm{where},i})$, $z^{\mathrm{pres},i} = 1$ for $i = 1 \dots n$ and h_θ^{dec} is the object decoder with parameters θ. It is composed of a *glimpse decoder* $f_\theta^{\mathrm{dec}} : \mathbf{g}_t^i \mapsto \mathbf{y}_t^i$, which constructs an image patch and a spatial transformer (ST, **Jaderberg2015**), which scales and shifts it according to $\mathbf{z}^{\mathrm{where}}$; see Figure 3.2.1 for details.

Inference Eslami2016 use a sequential inference algorithm, where latent variables are inferred one at a time; see Figure 3.3.1. The number of inference steps n is given by $z^{\mathrm{pres},1:n+1}$, a random vector of n ones followed by a zero. The $\mathbf{z}^i$ are sampled sequentially from

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = q_\phi\Big(z^{\mathrm{pres},n+1} = 0 \mid \mathbf{z}^{\mathrm{w},1:n}, \mathbf{x}\Big) \prod_{i=1}^{n} q_\phi\Big(\mathbf{z}^{\mathrm{w},i}, z^{\mathrm{pres},i} = 1 \mid \mathbf{z}^{1:i-1}, \mathbf{x}\Big), \tag{5.2}$$

where q_ϕ is implemented as a neural network with parameters ϕ. To implement explaining away, e.g. to avoid encoding the same object twice, it is vital to capture the dependency of $\mathbf{z}^{\mathrm{w},i}$ and $z^{\mathrm{pres},i}$ on $\mathbf{z}^{1:i-1}$ and $\mathbf{x}$. This is done using a RNN R_ϕ with hidden state $\boldsymbol{h}^i$, namely: $\boldsymbol{\omega}^i, \boldsymbol{h}^i = R_\phi(\mathbf{x}, \mathbf{z}^{i-1}, \boldsymbol{h}^{i-1})$. The outputs $\boldsymbol{\omega}^i$, which are computed iteratively and depend on the previous latent variables (*cf.* Algorithm 3), parametrise $q_\phi(\mathbf{z}^{\mathrm{w},i}, \mathbf{z}^{\mathrm{pres},i} \mid \mathbf{z}^{1:i-1}, \mathbf{x})$. For simplicity the latter is assumed to factorise such that $q_\phi(\mathbf{z}^{\mathrm{w}}, \mathbf{z}^{\mathrm{pres}} \mid \mathbf{z}^{1:i-1}, \mathbf{x}) = q_\phi(z^{\mathrm{pres},n+1} = 0 \mid \boldsymbol{\omega}^{n+1}) \prod_{i=1}^{n} q_\phi(\mathbf{z}^{\mathrm{w},i} \mid \boldsymbol{\omega}^i) q_\phi(z^{\mathrm{pres},i} = 1 \mid \boldsymbol{\omega}^i)$.

5.3 Sequential Attend-Infer-Repeat

While capable of decomposing a scene into objects, AIR only describes single images. Should we want a similar decomposition of an image sequence, it would be desirable

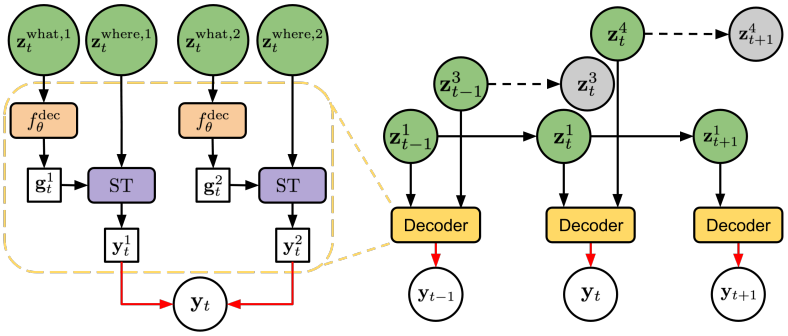


Figure 5.2.1: *Left*: Generation in AIR. The image mean $\mathbf{y}_t$ is generated by first using the *glimpse decoder* f_θ^{dec} to map the *what* variables into glimpses $\mathbf{g}_t$, transforming them with the *spatial transformer* ST according to the *where* variables and summing up the results. *Right*: Generation in SQAIR.

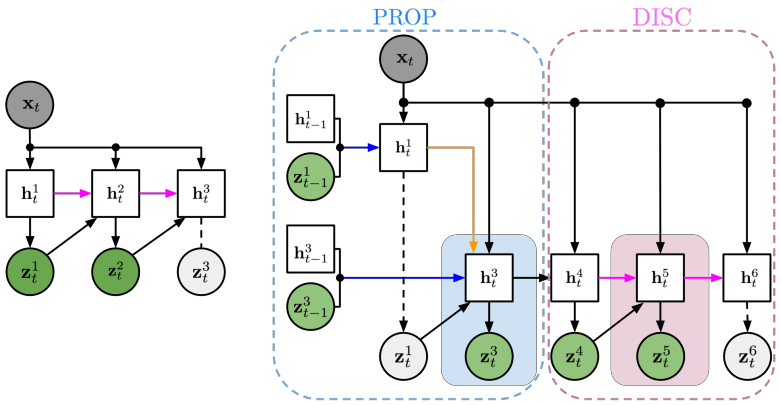


Figure 5.3.1: *Left*: Inference in AIR. The pink RNN attends to the image sequentially and produces one latent variable $\mathbf{z}_t^i$ at a time. Here, it decides that two latent variables are enough to explain the image and $\mathbf{z}_t^3$ is not generated. *Right*: Inference in SQAIR starts with the PROP phase. PROP iterates over latent variables from the previous time-step $t-1$ and updates them based on the new observation $\mathbf{x}_t$. The blue RNN runs forward in time to update the hidden state of each object, to model its change in appearance and location throughout time. The orange RNN runs across all current objects and models the relations between different objects. Here, when attending to $\mathbf{z}_{t-1}^1$, it decides that the corresponding object has disappeared from the frame and *forgets* it. Next, the DISC phase detects new objects as in AIR, but in SQAIR it is also conditioned on the results of PROP, to prevent rediscovering objects. See Figure 3.3.2 for details of the colored RNNs.

to do so in a temporally consistent manner. For example, we might want to detect objects of the scene as well as infer dynamics and track identities of any persistent objects. Thus, we introduce Sequential Attend, Infer, Repeat (SQAIR), whereby AIR is augmented with a state-space model (SSM) to achieve temporal consistency in the generated images of the sequence. The resulting probabilistic model is composed of two parts: Discovery (DISC), which is responsible for detecting (or introducing, in the case of the generation) new objects at every time-step (essentially equivalent to AIR), and Propagation (PROP), responsible for updating (or forgetting) latent variables from the previous time-step given the new observation (image), effectively implementing the temporal SSM. We now formally introduce SQAIR by first describing its generative model and then the inference network.

Generative Model The model assumes that at every-time step, objects are first propagated from the previous time-step (PROP). Then, new objects are introduced (DISC). Let $t \in \mathbb{N}$ be the current time-step. Let $\mathcal{P}_t$ be the set of objects propagated from the previous time-step and let $\mathcal{D}_t$ be the set of objects discovered at the current time-step, and let $\mathcal{O}_t = \mathcal{P}_t \cup \mathcal{D}_t$ be the set of all objects present at time-step t. Consequently, at every time step, the model retains a set of latent variables $\mathbf{z}_t^{\mathcal{P}_t} = \{\mathbf{z}_t^i\}_{i \in \mathcal{P}_t}$, and generates a set of new latent variables $\mathbf{z}_t^{\mathcal{D}_t} = \{\mathbf{z}_t^i\}_{i \in \mathcal{D}_t}$. Together they form $\mathbf{z}_t := [\mathbf{z}_t^{\mathcal{P}_t}, \mathbf{z}_t^{\mathcal{D}_t}]$, where the representation of the i^{th} object $\mathbf{z}_t^i := [\mathbf{z}_t^{\text{what},i}, \mathbf{z}_t^{\text{where},i}, z_t^{\text{pres},i}]$ is composed of three components (as in AIR): $\mathbf{z}_t^{\text{what},i}$ and $\mathbf{z}_t^{\text{where},i}$ are real vector-valued variables representing appearance and location of the object, respectively. $z_t^{\text{pres},i}$ is a binary variable representing whether the object is present at the given time-step or not.

At the first time-step ($t = 1$) there are no objects to propagate, so we sample D_1, the number of objects at $t = 1$, from the discovery prior $p^D(D_1)$. Then for each object $i \in \mathcal{D}_t$, we sample latent variables $\mathbf{z}_t^{\text{what},i}, \mathbf{z}_t^{\text{where},i}$ from $p^D(z_1^i \mid D_1)$. At time $t = 2$, the *propagation* step models which objects from $t = 1$ are propagated to $t = 2$, and which objects disappear from the frame, using the binary random variable $(z_t^{\text{pres},i})_{i \in \mathcal{P}_t}$. The *discovery* step at $t = 2$ models new objects that enter the frame, with a similar procedure to $t = 1$: sample D_2 (which depends on $\mathbf{z}_2^{\mathcal{P}_2}$) then

sample $(\mathbf{z}_2^{\text{what},i}, \mathbf{z}_2^{\text{where},i})_{i\in\mathcal{D}_2}$. This procedure of propagation and discovery recurs for $t = 2, \dots T$. Once the $\mathbf{z}_t$ have been formed, we may generate images $\mathbf{x}_t$ using the exact same generative distribution $p_\theta(\mathbf{x}_t \mid \mathbf{z}_t)$ as in AIR (*cf.* Equation (3.1), Fig. 3.2.1, and Algorithm 1). In full, the generative model is:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}, D_{1:T}) = p^D(D_1, \mathbf{z}_1^{\mathcal{D}_1}) \prod_{t=2}^{T} p^D(D_t, \mathbf{z}_t^{\mathcal{D}_t} | \mathbf{z}_t^{\mathcal{P}_t}) p^P(\mathbf{z}_t^{\mathcal{P}_t} | \mathbf{z}_{t-1}) p_\theta(\mathbf{x}_t | \mathbf{z}_t), \quad (5.3)$$

The *discovery prior* $p^D(D_t, \mathbf{z}_t^{\mathcal{D}_t} | \mathbf{z}_t^{\mathcal{P}_t})$ samples latent variables for new objects that enter the frame. The *propagation prior* $p^P(\mathbf{z}_t^{\mathcal{P}_t} | \mathbf{z}_{t-1})$ samples latent variables for objects that persist in the frame and removes latents of objects that disappear from the frame, thereby modelling dynamics and appearance changes. Both priors are learned during training. The exact forms of the priors are given in Section 3.B.

Inference Similarly to AIR, inference in SQAIR can capture the number of objects and the representation describing the location and appearance of each object that is necessary to explain every image in a sequence. As with generation, inference is divided into PROP and DISC. During PROP, the inference network achieves two tasks. Firstly, the latent variables from the previous time step are used to infer the current ones, modelling the change in location and appearance of the corresponding objects, thereby attaining temporal consistency. This is implemented by the *temporal* RNN R_ϕ^T, with hidden states $\boldsymbol{h}_t^T$ (recurs in t). Crucially, it does not access the current image directly, but uses the output of the *relation* RNN (*cf.* **Santoro2017**). The relation RNN takes relations between objects into account, thereby implementing the *explaining away* phenomenon; it is essential for capturing any interactions between objects as well as occlusion (or overlap, if one object is occluded by another). See Figure 3.4.4 for an example. These two RNNs together decide whether to retain or to forget objects that have been propagated from the previous time step. During DISC, the network infers further latent variables that are needed to describe any new objects that have entered the frame. All latent variables remaining after PROP and DISC are passed on to the next time step.

See Figures 3.3.1 and 3.3.2 for the inference network structure . The full variational posterior is defined as

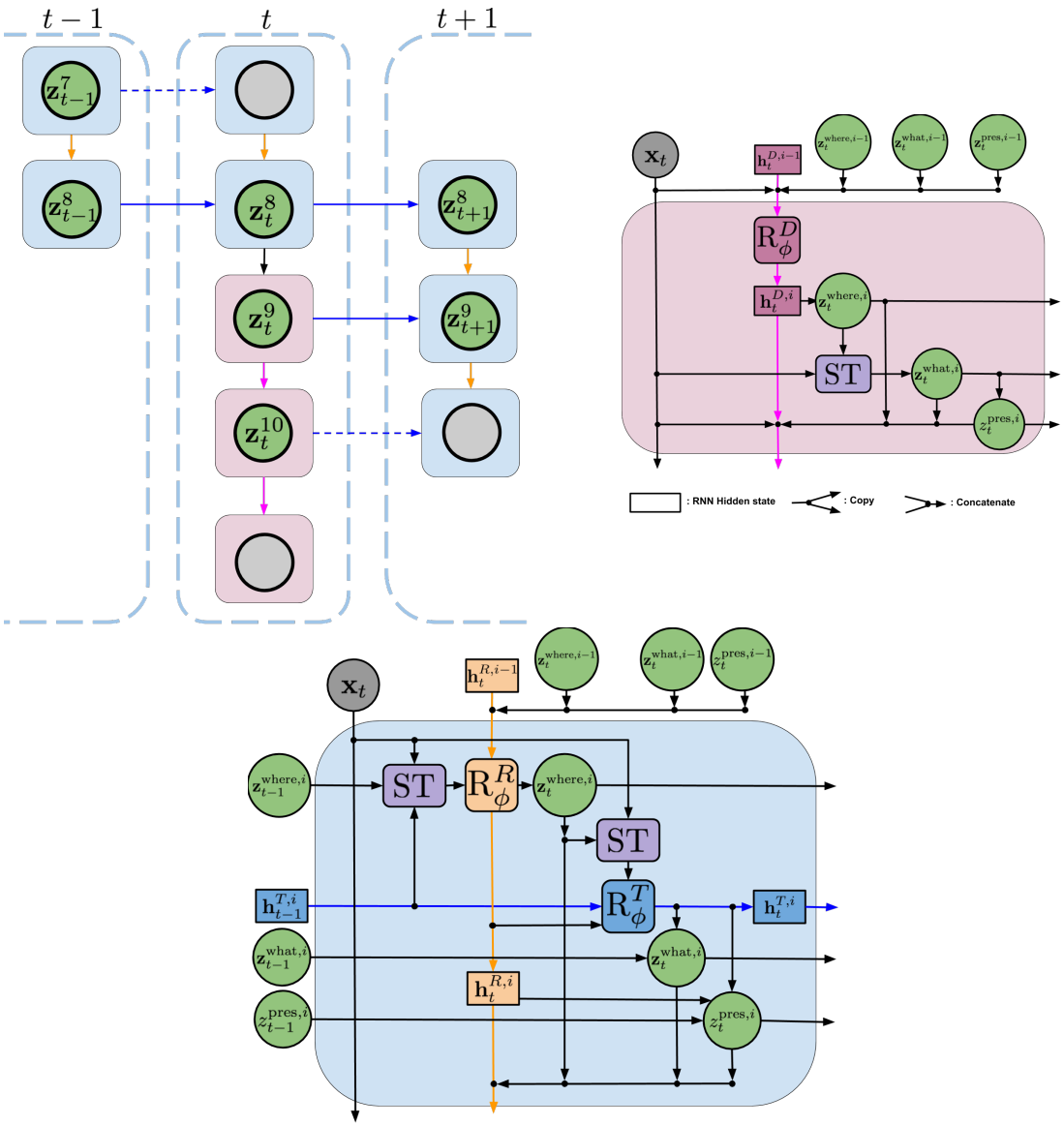


Figure 5.3.2: *Left*: Interaction between PROP and DISC in SQAIR. Firstly, objects are propagated to time t, and object $i = 7$ is dropped. Secondly, DISC tries to discover new objects. Here, it manages to find two objects: $i = 9$ and $i = 10$. The process recurs for all remaining time-steps. Blue arrows update the temporal hidden state, orange ones infer relations between objects, pink ones correspond to discovery. *Bottom*: Information flow in a single discovery block (*left*) and propagation block (*right*). In DISC we first predict *where* and extract a glimpse. We then predict *what* and *presence*. PROP starts with extracting a glimpse at a candidate location and updating *where*. Then it follows a procedure similar to DISC, but takes the respective latent variables from the previous time-step into account. It is approximately two times more computationally expensive than DISC. For details, see Algorithms 2 and 3 in Section 3.A.

$$q_\phi(D_{1:t}, \mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}) = \prod_{t=1}^{T} q_\phi^D\left(D_t, \mathbf{z}_t^{\mathcal{P}_t} \mid \mathbf{x}_t, \mathbf{z}_t^{\mathcal{P}_t}\right) \prod_{i \in \mathcal{O}_{t-1}} q_\phi^P\left(\mathbf{z}_t^i \mid \mathbf{z}_{t-1}^i, \boldsymbol{h}_t^{T,i}, \boldsymbol{h}_t^{R,i}\right). \quad (5.4)$$

Discovery, described by q_ϕ^D, is very similar to the full posterior of AIR, *cf.* Equation (3.2). The only difference is the conditioning on $\mathbf{z}_t^{\mathcal{P}_t}$, which allows for a different number of discovered objects at each time-step and also for objects explained by PROP not to be explained again. The second term, or q_ϕ^P, describes propagation. The detailed structures of q_ϕ^D and q_ϕ^P are shown in Figure 3.3.2, while all the pertinent algorithms and equations can be found in Sections 3.A and 3.C, respectively.

Learning We train SQAIR as an IWAE of **Burda2016**. Specifically, we maximise the importance-weighted evidence lower-bound $\mathcal{L}_{\text{IWAE}}$, namely

$$\mathcal{L}_{\text{IWAE}} = \mathbb{E}_{\mathbf{x}_{1:T} \sim p_{\text{data}}(\mathbf{x}_{1:T})}\left[\mathbb{E}_q\left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T})}{q_\phi(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T})}\right]\right]. \quad (5.5)$$

To optimise the above, we use RMSPROP, $K = 5$ and batch size of 32. We use the VIMCO gradient estimator of **Mnih2016** to backpropagate through the discrete latent variables z^{pres}, and use reparameterisation for the continuous ones (**kingma2013auto**). We also tried to use NVIL of **Mnih2014** as in the original work on AIR, but found it very sensitive to hyper-parameters, fragile and generally under-performing.

5.4 Experiments

We evaluate SQAIR on two datasets. Firstly, we perform an extensive evaluation on moving MNIST digits, where we show that it can learn to reliably detect, track and generate moving digits (Section 3.4.1). Moreover, we show that SQAIR can simulate moving objects into the future — an outcome it has not been trained for. We also study the utility of learned representations for a downstream task. Secondly, we apply SQAIR to real-world pedestrian CCTV data from static cameras (*DukeMTMC*, **ristani2016performance**), where we perform background subtraction as pre-processing. In this experiment, we show that SQAIR learns to detect, track, predict and generate walking pedestrians without human supervision.

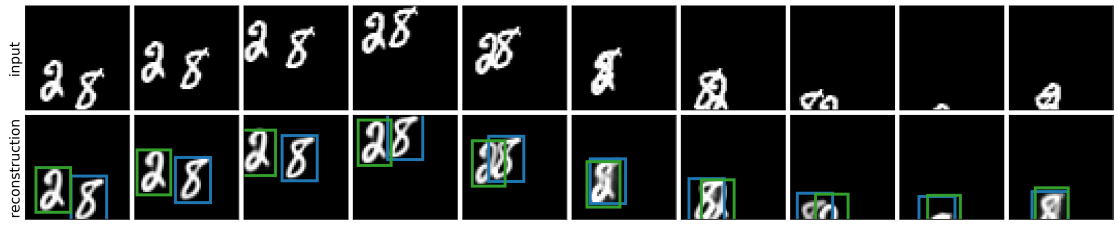


Figure 5.4.1: Input images (top) and SQAIR reconstructions with marked glimpse locations (bottom). For more examples, see Figure 3.H.1 in Section 3.H.

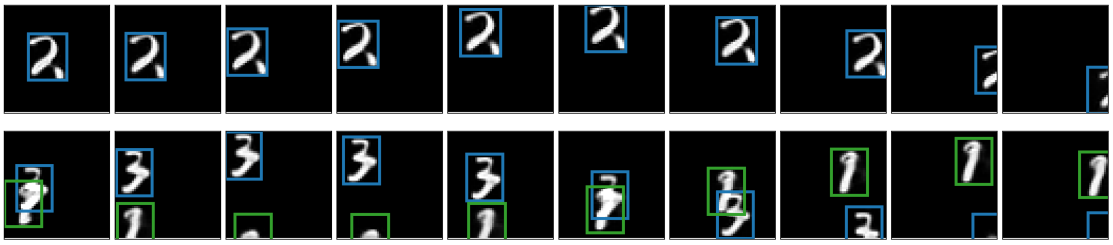


Figure 5.4.2: Samples from SQAIR. Both motion and appearance are consistent through time, thanks to the propagation part of the model. For more examples, see Figure 3.H.3 in Section 3.H.

5.4.1 Moving multi-mnist

The dataset consists of sequences of length 10 of multiple moving MNIST digits. All images are of size 50×50 and there are zero, one or two digits in every frame (with equal probability). Sequences are generated such that no objects overlap in the first frame, and all objects are present through the sequence; the digits can move out of the frame, but always come back. See Section 3.F for an experiment on a harder version of this dataset. There are 60,000 training and 10,000 testing sequences created from the respective MNIST datasets. We train two variants of SQAIR: the MLP-SQAIR uses only fully-connected networks, while the CONV-SQAIR replaces the

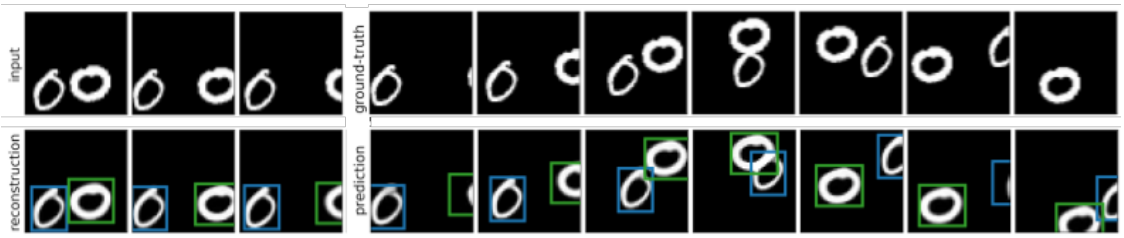


Figure 5.4.3: The first three frames are input to SQAIR, which generated the rest conditional on the first frames.

Figure 5.4.4: Inputs, reconstructions with marked glimpse locations and reconstructed glimpses for AIR (left) and SQAIR (right). SQAIR can model partially visible and heavily overlapping objects by aggregating temporal information.

	$\log p_\theta(\mathbf{x}_{1:T})$	$\log p_\theta(\mathbf{x}_{1:T} \mid \mathbf{z}_{1:T})$	$\mathrm{KL}(q_\phi \mid\mid p_\theta)$	Counting	Addition
CONV-SQAIR	**6784.8**	**6923.8**	**134.6**	0.9974	0.9990
MLP-SQAIR	6617.6	6786.5	164.5	**0.9986**	**0.9998**
MLP-AIR	6443.6	6830.6	352.6	0.9058	0.8644
CONV-VRNN	6561.9	6737.8	270.2	n/a	0.8536
MLP-VRNN	5959.3	6108.7	218.3	n/a	0.8059

Table 5.4.1: SQAIR achieves higher performance than the baselines across a range of metrics. The third column refers to the KL divergence between the approximate posterior and the prior. Counting refers to accuracy of the inferred number of objects present in the scene, while addition stands for the accuracy of a supervised digit addition experiment, where a classifier is trained on the learned latent representations of each frame.

networks used to encode images and glimpses with convolutional ones; it also uses a subpixel-convolution network as the glimpse decoder (**shi2016subpixel**). See Section 3.D for details of the model architectures and the training procedure.

We use AIR and VRNN (**Chung2015**) as baselines for comparison. VRNN can be thought of as a sequential VAE with an RNN as its deterministic backbone. Being similar to a VAE, its latent variables are not structured, nor easily interpretable. For a fair comparison, we control the latent dimensionality of VRNN and the number of learnable parameters. We provide implementation details in Section 3.D.3.

The quantitative analysis consists of comparing all models in terms of the marginal log-likelihood $\log p_\theta(\mathbf{x}_{1:T})$ evaluated as the $\mathcal{L}_{\mathrm{IWAE}}$ bound with $K = 1000$ particles, reconstruction quality evaluated as a single-sample approximation of $\mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}_{1:T} \mid \mathbf{z}_{1:T})]$ and the KL-divergence between the approximate posterior and the prior (Table 3.4.1). Additionally, we measure the accuracy of the number of objects modelled by SQAIR and AIR. SQAIR achieves superior performance

across a range of metrics — its convolutional variant outperforms both AIR and the corresponding VRNN in terms of model evidence and reconstruction performance. The KL divergence for SQAIR is almost twice as low as for VRNN and by a yet larger factor for AIR. We can interpret KL values as an indicator of the ability to compress, and we can treat SQAIR/AIR type of scheme as a version of run-length encoding. While VRNN has to use information to explicitly describe every part of the image, even if some parts are empty, SQAIR can explicitly allocate content information ($\mathbf{z}^{\text{what}}$) to specific parts of the image (indicated by $\mathbf{z}^{\text{where}}$). AIR exhibits the highest values of KL, but this is due to encoding every frame of the sequence independently — its prior cannot take *what* and *where* at the previous time-step into account, hence higher KL. The fifth column of Table 3.4.1 details the object counting accuracy, that is indicative of the quality of the approximate posterior. It is measured as the sum of z_t^{pres} for a given frame against the true number of objects in that frame. As there is no z^{pres} for VRNN no score is provided. Perhaps surprisingly, this metric is much higher for SQAIR than for AIR. This is because AIR mistakenly infers overlapping objects as a single object. Since SQAIR can incorporate temporal information, it does not exhibit this failure mode (*cf.* Figure 3.4.4). Next, we gauge the utility of the learnt representations by using them to determine the sum of the digits present in the image (Table 3.4.1, column six). To do so, we train a 19-way classifier (mapping from any combination of up to two digits in the range $[0, 9]$ to their sum) on the extracted representations and use the summed labels of digits present in the frame as the target. Section 3.D contains details of the experiment. SQAIR significantly outperforms AIR and both variants of VRNN on this tasks. VRNN under-performs due to the inability of disentangling overlapping objects, while both VRNN and AIR suffer from low temporal consistency of learned representations, see Section 3.H. Finally, we evaluate SQAIR qualitatively by analyzing reconstructions and samples produced by the model against reconstructions and samples from VRNN. We observe that samples and reconstructions from SQAIR are of better quality and, unlike VRNN, preserve motion and appearance consistently through time. See Section 3.H for direct comparison and additional examples. Furthermore,

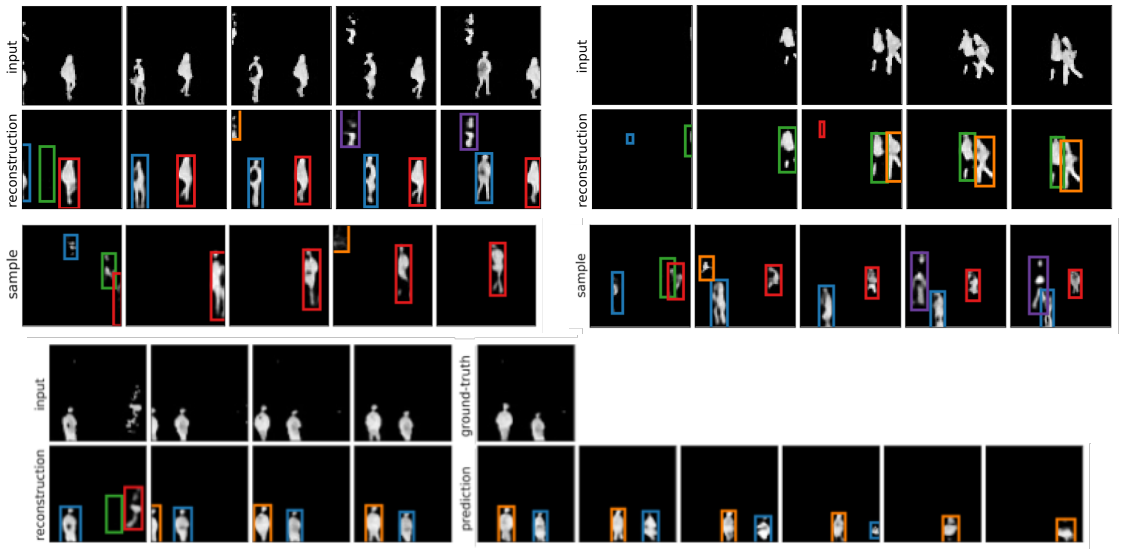


Figure 5.4.5: Inputs on the top, reconstructions in the second row, samples in the third row; rows four and five contain inputs and conditional generation: the first four frames in the last row are reconstructions, while the remaining ones are predicted by sampling from the prior. There is no ground-truth, since we used sequences of length five of training and validation.

we examine conditional generation, where we look at samples from the generative model of SQAIR conditioned on three images from a real sequence (see Figure 3.4.3). We see that the model can preserve appearance over time, and that the simulated objects follow similar trajectories, which hints at good learning of the motion model (see Section 3.H for more examples). Figure 3.4.4 shows reconstructions and corresponding glimpses of AIR and SQAIR. Unlike SQAIR, AIR is unable to recognize objects from partial observations, nor can it distinguish strongly overlapping objects (it treats them as a single object; columns five and six in the figure). We analyze failure cases of SQAIR in Section 3.G.

5.4.2 Generative Modelling of Walking Pedestrians

To evaluate the model in a more challenging, real-world setting, we turn to data from static CCTV cameras of the *DukeMTMC* dataset (**ristani2016performance**). As part of pre-precessing, we use standard background subtraction algorithms (**itseez2015opencv**). In this experiment, we use 3150 training and 350 validation sequences of length 5. For details of model architectures, training and data pre-

processing, see Section 3.E. We evaluate the model qualitatively by examining reconstructions, conditional samples (conditioned on the first four frames) and samples from the prior (Figure 3.4.5 and Section 3.I). We see that the model learns to reliably detect and track walking pedestrians, even when they are close to each other.

There are some spurious detections and re-detections of the same objects, which is mostly caused by imperfections of the background subtraction pipeline — backgrounds are often noisy and there are sudden appearance changes when a part of a person is treated as background in the pre-processing pipeline. The object counting accuracy in this experiment is 0.5712 on the validation dataset, and we noticed that it does increase with the size of the training set. We also had to use early stopping to prevent overfitting, and the model was trained for only 315k iterations ($>$ 1M for MNIST experiments). Hence, we conjecture that accuracy and marginal likelihood can be further improved by using a bigger dataset.

5.5 Related Work

Object Tracking There have been many approaches to modelling objects in images and videos. Object detection and tracking are typically learned in a supervised manner, where object bounding boxes and often additional labels are part of the training data. Single-object tracking commonly use Siamese networks, which can be seen as an RNN unrolled over two time-steps (**valmadre2017end**). Recently, **kosiorek2017hierch** used an RNN with an attention mechanism in the HART model to predict bounding boxes for single objects, while robustly modelling their motion and appearance. Multi-object tracking is typically attained by detecting objects and performing data association on bounding-boxes (**bewley2016sort**). **schulter2017deepnf** used an end-to-end supervised approach that detects objects and performs data association. In the unsupervised setting, where the training data consists of only images or videos, the dominant approach is to distill the inductive bias of spatial consistency into a discriminative model. **cho2015unsupervised** detect single objects and their parts in images, and **kwak2015unsupervised**; **xiao2016track** incorporate temporal consistency to

better track single objects. SQAIR is unsupervised and hence it does not rely on bounding boxes nor additional labels for training, while being able to learn arbitrary motion and appearance models similarly to HART (**kosiorek2017hierch**). At the same time, is inherently multi-object and performs data association implicitly (*cf.* Section 3.A). Unlike the other unsupervised approaches, temporal consistency is baked into the model structure of SQAIR and further enforced by lower KL divergence when an object is tracked.

Video Prediction Many works on video prediction learn a deterministic model conditioned on the current frame to predict the future ones (**ranzato2014video**; **srivastava2015unsupervised**). Since these models do not model uncertainty in the prediction, they can suffer from the multiple futures problem — since perfect prediction is impossible, the model produces blurry predictions which are a mean of possible outcomes. This is addressed in stochastic latent variable models trained using variational inference to generate multiple plausible videos given a sequence of images (**babaeizadeh2017stochastic**; **denton2018stochastic**). Unlike SQAIR, these approaches do not model objects or their positions explicitly, thus the representations they learn are of limited interpretability.

Learning Decomposed Representations of Images and Videos Learning decomposed representations of object appearance and position lies at the heart of our model. This problem can be also seen as perceptual grouping, which involves modelling pixels as spatial mixtures of entities. **Greff2016tagger** and **Greff2017neuralem** learn to decompose images into separate entities by iterative refinement of spatial clusters using either learned updates or the Expectation Maximization algorithm; **Ilin2017recurrentln** and **Steenkiste2018relationalnem** extend these approaches to videos, achieving very similar results to SQAIR. Perhaps the most similar work to ours is the concurrently developed model of **Hsieh2018ddpae**. The above approaches rely on iterative inference procedures, but do not exhibit the object-counting behaviour of SQAIR. For this reason, their computational complexities are proportional to the predefined maximum number

of objects, while SQAIR can be more computationally efficient by adapting to the number of objects currently present in an image.

Another interesting line of work is the GAN-based unsupervised video generation that decomposes motion and content (**tulyakov2017mocogan**; **denton2017unsupervised**). These methods learn interpretable features of content and motion, but deal only with single objects and do not explicitly model their locations. Nonetheless, adversarial approaches to learning structured probabilistic models of objects offer a plausible alternative direction of research.

Bayesian Nonparametric Models To the best of our knowledge, **neiswanger2012unsupervised** is the only known approach that models pixels belonging to a variable number of objects in a video together with their locations in the generative sense. This work uses a Bayesian nonparametric (BNP) model, which relies on mixtures of Dirichlet processes to cluster pixels belonging to an object. However, the choice of the model necessitates complex inference algorithms involving Gibbs sampling and Sequential Monte Carlo, to the extent that any sensible approximation of the marginal likelihood is infeasible. It also uses a fixed likelihood function, while ours is learnable.

The object appearance-persistence-disappearance model in SQAIR is reminiscent of the Markov Indian buffet process (MIBP) of **Gael2009**, another BNP model. MIBP was used as a model for blind source separation, where multiple sources contribute toward an audio signal, and can appear, persist, disappear and reappear independently. The prior in SQAIR is similar, but the crucial differences are that SQAIR combines the BNP prior with flexible neural network models for the dynamics and likelihood, as well as variational learning via amortized inference. The interface between deep learning and BNP, and graphical models in general, remains a fertile area of research.

5.6 Discussion

In this paper we proposed SQAIR, a probabilistic model that extends AIR to image sequences, and thereby achieves temporally consistent reconstructions and samples. In doing so, we enhanced AIR's capability of disentangling overlapping objects and identifying partially observed objects.

This work continues the thread of **Greff2017neuralem**, **Steenkiste2018relationalnem** and, together with **Hsieh2018ddpae**, presents unsupervised object detection & tracking with learnable likelihoods by the means of generative modelling of objects. In particular, our work is the first one to explicitly model object presence, appearance and location through time. Being a generative model, SQAIR can be used for conditional generation, where it can extrapolate sequences into the future. As such, it would be interesting to use it in a reinforcement learning setting in conjunction with Imagination-Augmented Agents (**weber2017imagination**) or more generally as a world model (**ha2018worldm**), especially for settings with simple backgrounds, e. g., games like Montezuma's Revenge or Pacman.

The framework offers various avenues of further research; SQAIR leads to interpretable representations, but the interpretability of *what* variables can be further enhanced by using alternative objectives that disentangle factors of variation in the objects (**kim2018disentangling**). Moreover, in its current state, SQAIR can work only with simple backgrounds and static cameras. In future work, we would like to address this shortcoming, as well as speed up the sequential inference process whose complexity is linear in the number of objects. The generative model, which currently assumes additive image composition, can be further improved by e. g., autoregressive modelling (**oord2016cond**). It can lead to higher fidelity of the model and improved handling of occluded objects. Finally, the SQAIR model is very complex, and it would be useful to perform a series of ablation studies to further investigate the roles of different components.

Acknowledgements

We would like to thank Ali Eslami for his help in implementing AIR, Alex Bewley and Martin Engelcke for discussions and valuable insights and anonymous reviewers for their constructive feedback. Additionally, we acknowledge that HK and YWT's research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 617071.

Appendix

5.A Algorithms

Image generation, described by Algorithm 1, is exactly the same for SQAIR and AIR. Algorithms 2 and 3 describe inference in SQAIR. Note that DISC is equivalent to AIR if no latent variables are present in the inputs.

If a function has multiple inputs and if not stated otherwise, all the inputs are concatenated and linearly projected into some fixed-dimensional space, e. g., Lines 9 and 15 in Algorithm 2. Spatial Transformer (ST, e. g., Line 7 in Algorithm 2) has no learnable parameters: it samples a uniform grid of points from an image $\mathbf{x}$, where the grid is transformed according to parameters $\mathbf{z}^{\text{where}}$. f_ϕ^1 is implemented as a perceptron with a single hidden layer. Statistics of q^P and q^D are a result of applying a two-layer multilayer perceptron (MLP) to their respective conditioning sets. Different distributions q do not share parameters of their MLPs. The *glimpse encoder* $\text{h}_\phi^{\text{glimpse}}$ (Lines 8 and 12 in Algorithm 2 and Line 12 in Algorithm 3; they share parameters) and the *image encoder* $\text{h}_\phi^{\text{enc}}$ (Line 3 in Algorithm 3) are implemented as two-layer MLPs or CNNs, depending on the experiment (see Sections 3.D and 3.E for details).

One of the important details of PROP is the proposal glimpse extracted in lines Lines 6 and 7 of Algorithm 2. It has a dual purpose. Firstly, it acts as an information bottleneck in PROP, limiting the flow of information from the current observation $\mathbf{x}_t$ to the updated latent variables $\mathbf{z}_t$. Secondly, even though the information is limited, it can still provide a high-resolution view of the object corresponding to the currently updated latent variable, *given* that the location of the proposal glimpse correctly predicts motion of this object. Initially, our implementation used encoding of the raw observation ($\text{h}_\phi^{\text{enc}}(\mathbf{x}_t)$, similarly to Line 3 in Algorithm 3) as an input to the relation-RNN (Line 9 in Algorithm 2). We have also experimented

with other bottlenecks: (1) low resolution image as an input to the image encoder and (2) a low-dimensional projection of the image encoding before the relation-RNN. Both approaches have led to *ID swaps*, where the order of explaining objects were sometimes swapped for different frames of the sequence (see Figure 3.G.1 in Section 3.G for an example). Using encoded proposal glimpse extracted from a predicted location has solved this issue.

To condition DISC on propagated latent variables (Line 4 in Algorithm 3), we encode the latter by using a two-layer MLP similarly to **zaheer2017deeps**,

$$\mathbf{l}_t = \sum_{i \in \mathcal{P}_t} \mathrm{MLP}\left(\mathbf{z}_t^{\mathrm{what},i}, \mathbf{z}_t^{\mathrm{where},i}\right). \tag{5.6}$$

Note that other encoding schemes are possible, though we have experimented only with this one.

Algorithm 1: Image Generation

Input : $\mathbf{z}_t^{\mathrm{what}}, \mathbf{z}_t^{\mathrm{where}}$ - latent variables from the current time-step.

1 $\mathcal{O}_t = \mathrm{indices}\left(\mathbf{z}_t^{\mathrm{what}}\right)$ `// Indices of all present latent variables.`

2 $\mathbf{y}_t^0 = \mathbf{0}$

3 **for** $i \in \mathcal{O}_t$ **do**

4 $\quad \mathbf{y}_t^{\mathrm{att},i} = \mathrm{f}_\theta^{\mathrm{dec}}\left(\mathbf{z}_t^{\mathrm{what},i}\right)$ `// Decode the glimpse.`

5 $\quad \mathbf{y}_t^i = \mathbf{y}_t^{i-1} + \mathrm{ST}^{-1}\left(\mathbf{y}_t^{\mathrm{att},i}, \mathbf{z}_t^{\mathrm{where},i}\right)$

6 $\hat{\mathbf{x}}_t \sim \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{y}_n, \sigma_x^2 \boldsymbol{I}\right)$

Output : $\hat{\boldsymbol{x}}$

Algorithm 2: Inference for Propagation

Input : $\boldsymbol{x}_t$ - image at the current time-step,
$\mathbf{z}_{t-1}^{\text{what}}, \mathbf{z}_{t-1}^{\text{where}}, \mathbf{z}_{t-1}^{\text{pres}}$ - latent variables from the previous time-step
$\boldsymbol{h}_{t-1}^{T}$ - hidden states from the previous time-step.

$\boldsymbol{h}_t^{R,0}, \boldsymbol{z}_t^{\text{what},0}, \boldsymbol{z}_t^{\text{where},0} = \text{initialize}()$
$j = 0$ `// Index of the object processed in the last iteration.`
for $i \in \mathcal{O}_{t-1}$ **do**
  if $z_{t-1}^{\text{pres},i} == 0$ **then**
    continue
  $\hat{\mathbf{z}}_t^{\text{where},i} = \text{f}_\phi^1\left(\mathbf{z}_{t-1}^{\text{where},i}, \boldsymbol{h}_t^{T,i}\right)$ `// Proposal location.`
  $\hat{\mathbf{g}}_t^i = \text{ST}\left(\mathbf{x}_t, \hat{\mathbf{z}}_t^{\text{where},i}\right)$ `// Extract a glimpse from a proposal location.`
  $\hat{\mathbf{e}}_t^i = \text{h}_\phi^{\text{glimpse}}\left(\hat{\mathbf{g}}_t^i\right)$ `// Encode the proposal glimpse.`
  $\mathbf{w}_t^{R,i}, \boldsymbol{h}_t^{R,i} = \text{R}_\phi^R\left(\hat{\mathbf{e}}_t^i, \mathbf{z}_{t-1}^{\text{what},i}, \mathbf{z}_{t-1}^{\text{where},i}, \boldsymbol{h}_{t-1}^{T,i}, \boldsymbol{h}_t^{R,j}, \mathbf{z}_t^{\text{what},j}, \mathbf{z}_t^{\text{where},j}\right)$ `// Relational state, see Equation (3.14).`
  $\mathbf{z}_t^{\text{where},i} \sim q_\phi^P\left(\boldsymbol{z}^{\text{where}} \mid \boldsymbol{z}_{t-1}^{\text{where},k}, \mathbf{w}_t^{R,i}\right)$
  $\mathbf{g}_t^i = \text{ST}\left(\mathbf{x}_t, \mathbf{z}_t^{\text{where},i}\right)$ `// Extract the final glimpse.`
  $\mathbf{e}_t^i = \text{h}_\phi^{\text{glimpse}}\left(\mathbf{g}_t^i\right)$ `// Encode the final glimpse.`
  $\mathbf{w}_t^{T,i}, \boldsymbol{h}_t^{T,i} = \text{R}_\phi^T\left(\mathbf{e}_t^i, \mathbf{z}_t^{\text{where},i}, \boldsymbol{h}_{t-1}^{T,i}, \boldsymbol{h}_t^{R,i}\right)$ `// Temporal state, see Equation (3.15).`
  $\mathbf{z}_t^{\text{what},i} \sim q_\phi^P\left(\mathbf{z}^{\text{what}} \mid \mathbf{e}_t^i, \mathbf{z}_{t-1}^{\text{what},i}, \mathbf{w}_t^{R,i}, \mathbf{w}_t^{T,i}\right)$
  $z_t^{\text{pres},i} \sim q_\phi^P\left(z^{\text{pres}} \mid z_{t-1}^{\text{pres},i}, \mathbf{z}_t^{\text{what},i}, \mathbf{z}_t^{\text{where},i}, \mathbf{w}_t^{R,i}, \mathbf{w}_t^{T,i}\right)$ `// Equation (3.13).`
  $j = i$

Output : $\mathbf{z}_t^{\text{what},\mathcal{P}_t}, \mathbf{z}_t^{\text{where},\mathcal{P}_t}, \mathbf{z}_t^{\text{pres},\mathcal{P}_t}$

5.B Details for the Generative Model of SQAIR

In implementation, we upper bound the number of objects at any given time by N. In detail, the discovery prior is given by

$$p^D\left(D_t, \mathbf{z}_t^{\mathcal{D}_t} \mid \mathbf{z}_t^{\mathcal{P}_t}\right) = p^D(D_t \mid P_t) \prod_{i \in \mathcal{D}_t} p^D(\mathbf{z}_t^{\text{what},i}) p^D(\mathbf{z}_t^{\text{where},i}) \delta_1(z_t^{\text{pres},i}), \tag{5.7}$$

$$p^D(D_t \mid P_t) = \text{Categorical}\left(D_t; N - P_t, p_\theta(P_t)\right), \tag{5.8}$$

where $\delta_x(\cdot)$ is the delta function at x, $\text{Categorical}(k; K, p)$ implies $k \in \{0, 1, \ldots, K\}$ with probabilities $p_0, p_1, \ldots, p_K$ and $p^D(\mathbf{z}_t^{\text{what},i}), p^D(\mathbf{z}_t^{\text{where},i})$ are fixed isotropic Gaus-

Algorithm 3: Inference for Discovery

Input : $\boldsymbol{x}_t$ - image at the current time-step,
$\mathbf{z}_t^{\mathcal{P}_t}$ - propagated latent variables for the current time-step,
N - maximum number of inference steps for discovery.

$\boldsymbol{h}_t^{D,0}, \boldsymbol{z}_t^{\text{what},0}, \boldsymbol{z}_t^{\text{where},0} = \text{initialize}()$
$j = \text{max\_index}\left(\mathbf{z}_t^{\mathcal{P}_t}\right)$ `// Maximum index among the propagated latent variables.`
$\mathbf{e}_t = \text{h}_\phi^{\text{enc}}(\mathbf{x}_t)$ `// Encode the image.`
$\mathbf{l}_t = \text{h}_\phi^{\text{enc}}\left(\mathbf{z}_t^{\text{what}}, \mathbf{z}_t^{\text{where}}, \mathbf{z}_t^{\text{pres}}\right)$ `// Encode latent variables.`
for $i \in [j+1, \ldots, j+N]$ **do**
  $\mathbf{w}_t^{D,i}, \boldsymbol{h}_t^{D,i} = \text{R}_\phi^D\left(\mathbf{e}_t, \mathbf{l}_t, \mathbf{z}_t^{\text{what},i-1}, \mathbf{z}_t^{\text{where},i-1}, \boldsymbol{h}_t^{D,i-1}\right)$
  $z_t^{\text{pres},i} \sim q_\phi^D\left(z^{\text{pres}} \mid \mathbf{w}_t^{D,i}\right)$
  if $z^{\text{pres},i} = 0$ **then**
    break
  $\mathbf{z}_t^{\text{where},i} \sim q_\phi^D\left(\mathbf{z}^{\text{where}} \mid \mathbf{w}_t^{D,i}\right)$
  $\mathbf{g}_t^i = \text{ST}\left(\mathbf{x}_t, \mathbf{z}_t^{\text{where},i}\right)$
  $\mathbf{e}_t^i = \text{h}_\phi^{\text{glimpse}}(\mathbf{g}_t^i)$ `// Encode the glimpse.`
  $\mathbf{z}_t^{\text{what},i} \sim q_\phi^D\left(\mathbf{z}_t^{\text{what}} \mid \mathbf{e}_t^i\right)$

Output : $\mathbf{z}_t^{\text{what},\mathcal{D}_t}, \mathbf{z}_t^{\text{where},\mathcal{D}_t}, \mathbf{z}_t^{\text{pres},\mathcal{D}_t}$

sians. The propagation prior is given by

$$p^P\left(\mathbf{z}_t^{\mathcal{P}_t} \mid \mathbf{z}_{t-1}\right) = \prod_{i \in \mathcal{P}_t} p^P\left(\mathbf{z}_t^{\text{pres},i} \mid \mathbf{z}_{t-1}^{\text{pres},i}, \boldsymbol{h}_{t-1}\right) p^P\left(\mathbf{z}_t^{\text{what},i} \mid \boldsymbol{h}_{t-1}\right) p^P\left(\mathbf{z}_t^{\text{where},i} \mid \boldsymbol{h}_{t-1}\right), \tag{5.9}$$

$$p^P\left(\mathbf{z}_t^{\text{pres},i} \mid \mathbf{z}_{t-1}^{\text{pres},i}, \boldsymbol{h}_{t-1}\right) = \text{Bernoulli}(z_t^{\text{pres},i}; f_\theta(\boldsymbol{h}_{t-1}))\delta_1(z_{t-1}^{\text{pres},i}), \tag{5.10}$$

with f_θ a scalar-valued function with range $[0, 1]$ and $p^P(\mathbf{z}_t^{\text{what},i}|\boldsymbol{h}_{t-1})$, $p^P(\mathbf{z}_t^{\text{where},i}|\boldsymbol{h}_{t-1})$ both factorised Gaussians parameterised by some function of $\boldsymbol{h}_{t-1}$.

5.C Details for the Inference of SQAIR

The propagation inference network q_ϕ^P is given as below,

$$q_\phi^P\left(\mathbf{z}_t^{\mathcal{P}_t} \mid \mathbf{x}_t, \mathbf{z}_{t-1}, \boldsymbol{h}_t^{T,\mathcal{P}_t}\right) = \prod_{i \in \mathcal{O}_{t-1}} q_\phi^P\left(\mathbf{z}_t^i \mid \mathbf{x}_t, \mathbf{z}_{t-1}^i, \boldsymbol{h}_t^{T,i}, \boldsymbol{h}_t^{R,i}\right), \tag{5.11}$$

with $\boldsymbol{h}_t^{R,i}$ the hidden state of the relation RNN (see Equation (3.14)). Its role is to capture information from the observation $\mathbf{x}_t$ as well as to model dependencies

between different objects. The propagation posterior for a single object can be expanded as follows,

$$\begin{aligned} &q_\phi^P\left(\mathbf{z}_t^i \mid \mathbf{x}_t, \mathbf{z}_{t-1}^i, \boldsymbol{h}_t^{T,i}, \boldsymbol{h}_t^{R,i}\right) = \\ &\quad q_\phi^P\left(\mathbf{z}_t^{\text{where},i} \mid \mathbf{z}_{t-1}^{\text{what},i}, \mathbf{z}_{t-1}^{\text{where},i}, \boldsymbol{h}_{t-1}^{T,i}, \boldsymbol{h}_t^{R,i}\right) \\ &\quad q_\phi^P\left(\mathbf{z}_t^{\text{what},i} \mid \mathbf{x}_t, \mathbf{z}_t^{\text{where},i}, \mathbf{z}_{t-1}^{\text{what},i}, \boldsymbol{h}_t^{T,i}, \boldsymbol{h}_t^{R,i}\right) \\ &\quad q_\phi^P\left(z_t^{\text{pres},i} \mid \mathbf{z}_t^{\text{what},i}, \mathbf{z}_t^{\text{where},i}, z_{t-1}^{\text{pres},i}, \boldsymbol{h}_t^{T,i}, \boldsymbol{h}_t^{R,i}\right). \end{aligned} \tag{5.12}$$

In the second line, we condition the object location $\mathbf{z}_t^{\text{where},i}$ on its previous appearance and location as well as its dynamics and relation with other objects. In the third line, current appearance $\mathbf{z}_t^{\text{what},i}$ is conditioned on the new location. Both $\mathbf{z}_t^{\text{where},i}$ and $\mathbf{z}_t^{\text{what},i}$ are modelled as factorised Gaussians. Finally, presence depends on the new appearance and location as well as the presence of the same object at the previous time-step. More specifically,

$$\begin{aligned} &q_\phi^P\left(z_t^{\text{pres},i} \mid \mathbf{z}_t^{\text{what},i}, \mathbf{z}_t^{\text{where},i}, z_{t-1}^{\text{pres},i}, \boldsymbol{h}_t^{T,i}, \boldsymbol{h}_t^{R,i}\right) \\ &\quad = \text{Bernoulli}\left(z_t^{\text{pres},i} \mid f_\phi\left(\mathbf{z}_t^{\text{what},i}, \mathbf{z}_t^{\text{where},i}, \boldsymbol{h}_t^{T,i}, \boldsymbol{h}_t^{R,i}\right)\right) \delta_1(z_{t-1}^{\text{pres},i}), \end{aligned} \tag{5.13}$$

where the second term is the delta distribution centered on the presence of this object at the previous time-step. If it was not there, it cannot be propagated. Let $j \in \{0, \ldots, i-1\}$ be the index of the most recent present object before object i. Hidden states are updated as follows,

$$\boldsymbol{h}_t^{R,i} = \mathrm{R}_\phi^R\left(\mathbf{x}_t, \mathbf{z}_{t-1}^{\text{what},i}, \mathbf{z}_{t-1}^{\text{where},i}, \boldsymbol{h}_{t-1}^{T,i}, \boldsymbol{h}_t^{R,i-1}, \mathbf{z}_t^{\text{what},j}, \mathbf{z}_t^{\text{where},j}\right), \tag{5.14}$$

$$\boldsymbol{h}_t^{T,i} = \mathrm{R}_\phi^T\left(\mathbf{x}_t, \mathbf{z}_t^{\text{where},i}, \boldsymbol{h}_{t-1}^{T,i}, \boldsymbol{h}_t^{R,i}\right), \tag{5.15}$$

where R_ϕ^T and R_ϕ^R are temporal and propagation RNNs, respectively. Note that in Eq. (3.14) the RNN does not have direct access to the image $\mathbf{x}_t$, but rather accesses it by extracting an attention glimpse at a proposal location, predicted from $\boldsymbol{h}_{t-1}^{T,i}$ and $\mathbf{z}_{t-1}^{\text{where},i}$. This might seem like a minor detail, but in practice structuring computation this way prevents ID swaps from occurring, *cf.* Section 3.G. For computational details, please see Algorithms 2 and 3 in Section 3.A.

5.D Details of the moving-mnist Experiments

5.D.1 Sqair and air Training Details

All models are trained by maximising the evidence lower bound (ELBO) $\mathcal{L}_{IWAE}$ (Equation (3.5)) with the RMSPROP optimizer (**tieleman2012rms**) with momentum equal to 0.9. We use the learning rate of 10^{-5} and decrease it to $\frac{1}{3} \cdot 10^{-5}$ after 400k and to 10^{-6} after 1000k training iterations. Models are trained for the maximum of $2 \cdot 10^6$ training iterations; we apply early stopping in case of overfitting. SQAIR models are trained with a curriculum of sequences of increasing length: we start with three time-steps, and increase by one time-step every 10^5 training steps until reaching the maximum length of 10. When training AIR, we treated all time-steps of a sequence as independent, and we trained it on all data (sequences of length ten, split into ten independent sequences of length one).

5.D.2 Sqair and air Model Architectures

All models use glimpse size of 20×20 and exponential linear unit (ELU) (**Clevert2015elu**) non-linearities for all layers except RNNs and output layers. MLP-SQAIR uses fully-connected layers for all networks. In both variants of SQAIR, the R^D_ϕ and R^R_ϕ RNNs are the vanilla RNNs. The propagation prior RNN and the temporal RNN R^T_ϕ use gated recurrent unit (GRU). AIR follows the same architecture as MLP-SQAIR. All fully-connected layers and RNNs in MLP-SQAIR and AIR have 256 units; they have 2.9M and 1.7M trainable parameters, respectively.

CONV-SQAIR differs from the MLP version in that it uses CNNs for the glimpse and image encoders and a subpixel-CNN (**shi2016subpixel**) for the glimpse decoder. All fully connected layers and RNNs have 128 units. The encoders share the CNN, which is followed by a single fully-connected layer (different for each encoder). The CNN has four convolutional layers with $[16, 32, 32, 64]$ features maps and strides of $[2, 2, 1, 1]$. The glimpse decoder is composed of two fully-connected layers with $[256, 800]$ hidden units, whose outputs are reshaped into 32 features maps of size 5×5, followed by a subpixel-CNN with three layers of $[32, 64, 64]$ feature maps and strides of $[1, 2, 2]$. All filters are of size 3×3. CONV-SQAIR has 2.6M trainable parameters.

We have experimented with different sizes of fully-connected layers and RNNs; we kept the size of all layers the same and altered it in increments of 32 units. Values greater than 256 for MLP-SQAIR and 128 for CONV-SQAIR resulted in overfitting. Models with as few as 32 units per layer (< 0.9M trainable parameters for MLP-SQAIR) displayed the same qualitative behaviour as reported models, but showed lower quantitative performance.

The output likelihood used in both SQAIR and AIR is Gaussian with a fixed standard deviation set to 0.3, as used by **Eslami2016**. We tried using a learnable scalar standard deviation, but decided not to report it due to unsable behaviour in the early stages of training. Typically, standard deviation would converge to a low value early in training, which leads to high penalties for reconstruction mistakes. In this regime, it is beneficial for the model to perform no inference steps (z^{pres} is always equal to zero), and the model never learns. Fixing standard deviation for the first 10k iterations and then learning it solves this issue, but it introduces unnecessary complexity into the training procedure.

5.D.3 Vrnn Implementation and Training Details

Our VRNN implementation is based on the implementation[3] of Filtering Variational Objectives (FIVO) by **maddison2017filtering**. We use an LSTM with hidden size J for the deterministic backbone of the VRNN. At time t, the LSTM receives $\psi^x(\mathbf{x}_{t-1})$ and $\psi^z(\mathbf{z}_{t-1})$ as input and outputs o_t, where ψ^x is a data feature extractor and ψ^z is a latent feature extractor. The output is mapped to the mean and standard deviation of the Gaussian prior $p_\theta(\mathbf{z}_t \mid \mathbf{x}_{t-1})$ by an MLP. The likelihood $p_\theta(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{x}_{t-1})$ is a Gaussian, with mean given by $\psi^{\text{dec}}(\psi^z(\mathbf{z}_t), o_t)$ and standard deviation fixed to be 0.3 as for SQAIR and AIR. The inference network $q_\phi(\mathbf{z}_t \mid \mathbf{z}_{t-1}, \mathbf{x}_t)$ is a Gaussian with mean and standard deviation given by the output of separate MLPs with inputs $[o_t, \psi^x(\mathbf{x}_t)]$.

All aforementioned MLPs use the same number of hidden units H and the same number of hidden layers L. The CONV-VRNN uses a CNN for ψ^x and a transposed CNN for ψ^{dec}. The MLP-VRNN uses an MLP with H' hidden units and L' hidden

[3] `https://github.com/tensorflow/models/tree/master/research/fivo`

	CONV-SQAIR	MLP-SQAIR	MLP-AIR	CONV-VRNN	MLP-VRNN
number of parameters	2.6M	2.9M	1.7M	2.6M	2.1M

Table 5.D.1: Number of trainable parameters for the reported models.

layers for both. ELU were used throughout as activations. The latent dimensionality was fixed to 165, which is the upper bound of the number of latent dimensions that can be used per time-step in SQAIR or AIR. Training was done by optimising the FIVO bound, which is known to be tighter than the IWAE bound for sequential latent variable models (**maddison2017filtering**). We also verified that this was the case with our models on the moving-MNIST data. We train with the RMSPROP optimizer with a learning rate of 10^{-5}, momentum equal to 0.9, and training until convergence of test FIVO bound.

For each of MLP-VRNN and CONV-VRNN, we experimented with three architectures: small/medium/large. We used $H{=}H'{=}J{=}128/256/512$ and $L{=}L'{=}2/3/4$ for MLP-VRNN, giving number of parameters of 1.2M/2.1M/9.8M. For CONV-VRNN, the number of features maps we used was $[32, 32, 64, 64]$, $[32, 32, 32, 64, 64, 64]$ and $[32, 32, 32, 64, 64, 64, 64, 64, 64]$, with strides of $[2, 2, 2, 2]$, $[1, 2, 1, 2, 1, 2]$ and $[1, 2, 1, 2, 1, 2, 1, 1, 1]$, all with 3×3 filters, $H{=}J{=}128/256/512$ and $L{=}1$, giving number of parameters of 0.8M/2.6M/6.1M. The largest convolutional encoder architecture is very similar to that in **gulrajani2016pixelvae** applied to MNIST.

We have chosen the medium-sized models for comparison with SQAIR due to overfitting encountered in larger models.

5.D.4 Addition Experiment

We perform the addition experiment by feeding latent representations extracted from the considered models into a 19-way classifier, as there are 19 possible outputs (addition of two digits between 0 and 9). The classifier is implemented as an MLP with two hidden layers with 256 ELU units each and a softmax output. For AIR and SQAIR, we use concatenated $\mathbf{z}^{\text{what}}$ variables multiplied by the corresponding z^{pres} variables, while for VRNN we use the whole 165-dimensional latent vector. We train the

model over 10^7 training iterations with the ADAM optimizer (**kingma2015adam**) with default parameters (in tensorflow).

5.E Details of the *DukeMTMC* Experiments

We take videos from cameras one, two, five, six and eight from the *DukeMTMC* dataset (**ristani2016performance**). As pre-processing, we invert colors and subtract backgrounds using standard OpenCV tools (**itseez2015opencv**), downsample to the resolution of 240×175, convert to gray-scale and randomly crop fragments of size 64×64. Finally, we generate 3500 sequences of length five such that the maximum number of objects present in any single frame is three and we split them into training and validation sets with the ratio of $9 : 1$.

We use the same training procedure as for the MNIST experiments. The only exception is the learning curriculum, which goes from three to five time-steps, since this is the maximum length of the sequences.

The reported model is similar to CONV-SQAIR. We set the glimpse size to 28×12 to account for the expected aspect ratio of pedestrians. Glimpse and image encoders share a CNN with $[16, 32, 64, 64]$ feature maps and strides of $[2, 2, 2, 1]$ followed by a fully-connected layer (different for each encoder). The glimpse decoder is implemented as a two-layer fully-connected network with 128 and 1344 units, whose outputs are reshaped into 64 feature maps of size 7×3, followed by a subpixel-CNN with two layers of $[64, 64]$ feature maps and strides of $[2, 2]$. All remaining fully-connected layers in the model have 128 units. The total number of trainable parameters is 3.5M.

5.F Harder multi-mnist Experiment

We created a version of the multi-MNIST dataset, where objects can appear or disappear at an arbitrary point in time. It differs from the dataset described in Section 3.4.1, where all digits are present throughout the sequence. All other dataset parameters are the same as in Section 3.4.1. Figure 3.F.1 shows an example sequence and MLP-SQAIR reconstructions with marked glimpse locations. The model

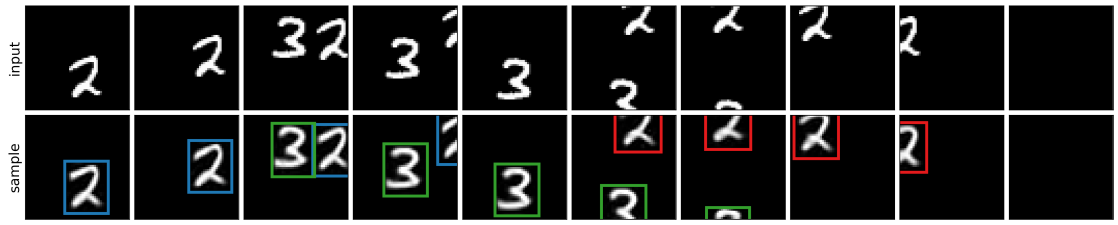


Figure 5.F.1: SQAIR trained on a harder version of moving-MNIST. Input images (top) and SQAIR reconstructions with marked glimpse locations (bottom)

has no trouble detecting new digits in the middle of the sequence and rediscovering a digit that was previously present.

5.G Failure cases of sqair

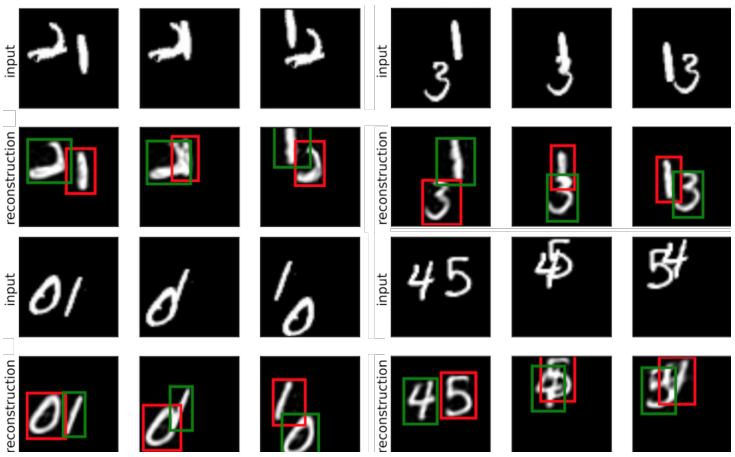


Figure 5.G.1: Examples of ID swaps in a version of SQAIR *without* proposal glimpse extraction in PROP (see Section 3.A for details). Bounding box colours correspond to object index (or its identity). When PROP is allowed the same access to the image as DISC, then it often prefers to ignore latent variables, which leads to swapped inference order.

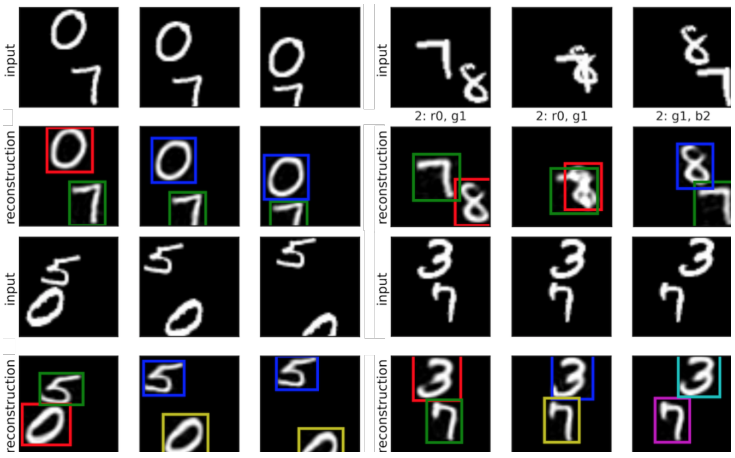


Figure 5.G.2: Examples of re-detections in MLP-SQAIR. Bounding box colours correspond to object identity, assigned to it upon discovery. In some training runs, SQAIR converges to a solution, where objects are re-detected in the second frame, and PROP starts tracking only in the third frame (left). Occasionally, an object can be re-detected after it has severely overlapped with another one (top right). Sometimes the model decides to use only DISC and repeatedly discovers all objects (bottom right). These failure mode seem to be mutually exclusive – they come from different training runs.

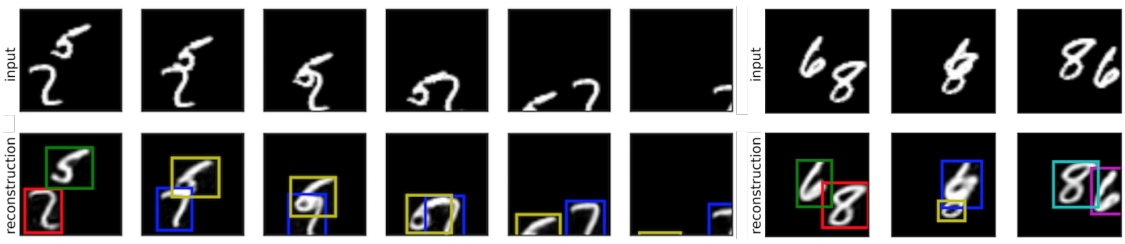
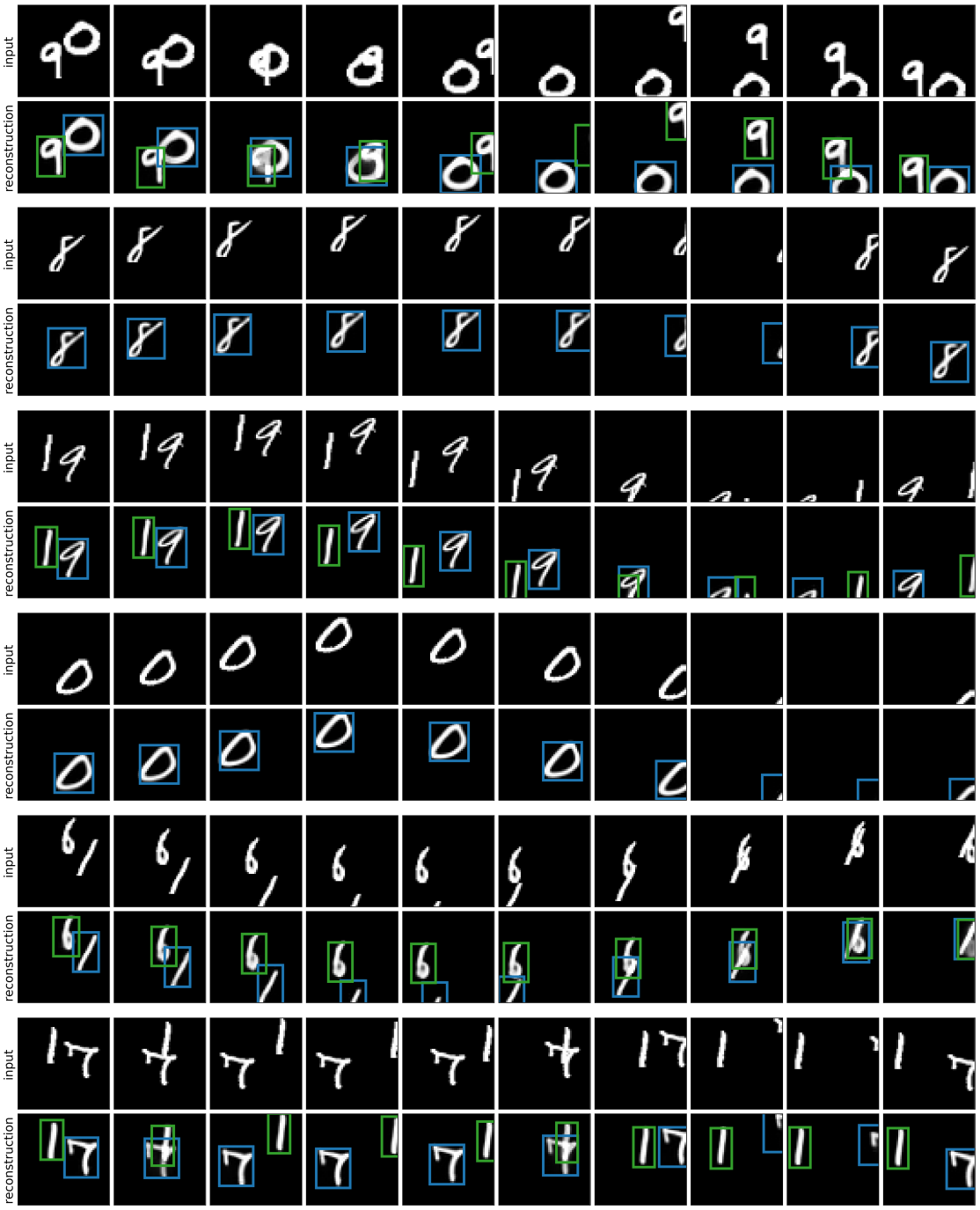


Figure 5.G.3: Two failed reconstructions of SQAIR. *Left*: SQAIR re-detects objects in the second time-step. Instead of 5 and 2, however, it reconstructs them as 6 and 7. Interestingly, reconstructions are consistent through the rest of the sequence. *Right:* At the second time-step, overlapping 6 and 8 are explained as 6 and a small 0. The model realizes its mistake in the third time-step, re-detects both digits and reconstructs them properly.

5.H Reconstruction and Samples from the Moving-MNIST Dataset

5.H.1 Reconstructions



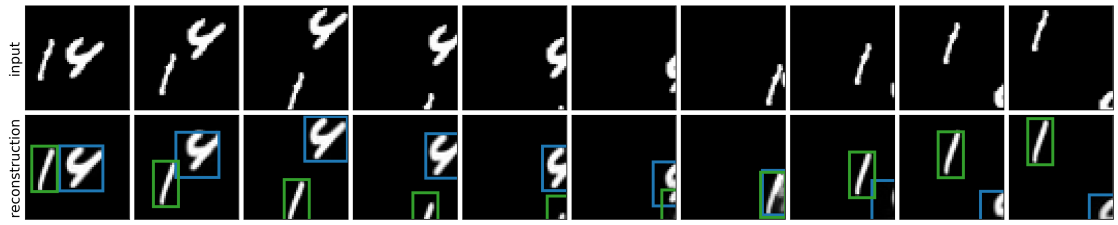


Figure 5.H.1: Sequences of input (first row) and SQAIR reconstructions with marked glimpse locations. Reconstructions are all temporally consistent.

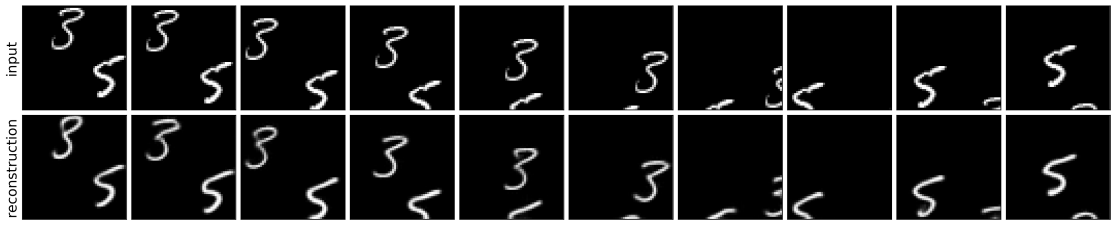


Figure 5.H.2: Sequences of input (first row) and CONV-VRNN reconstructions. They are not temporally consistent. The reconstruction at time $t = 1$ is typically of lower quality and often different than the rest of the sequence.

5.H.2 Samples

Figure 5.H.3: Samples from SQAIR. Both motion and appearance are temporally consistent. In the last sample, the model introduces the third object despite the fact that it has seen only up to two objects in training.

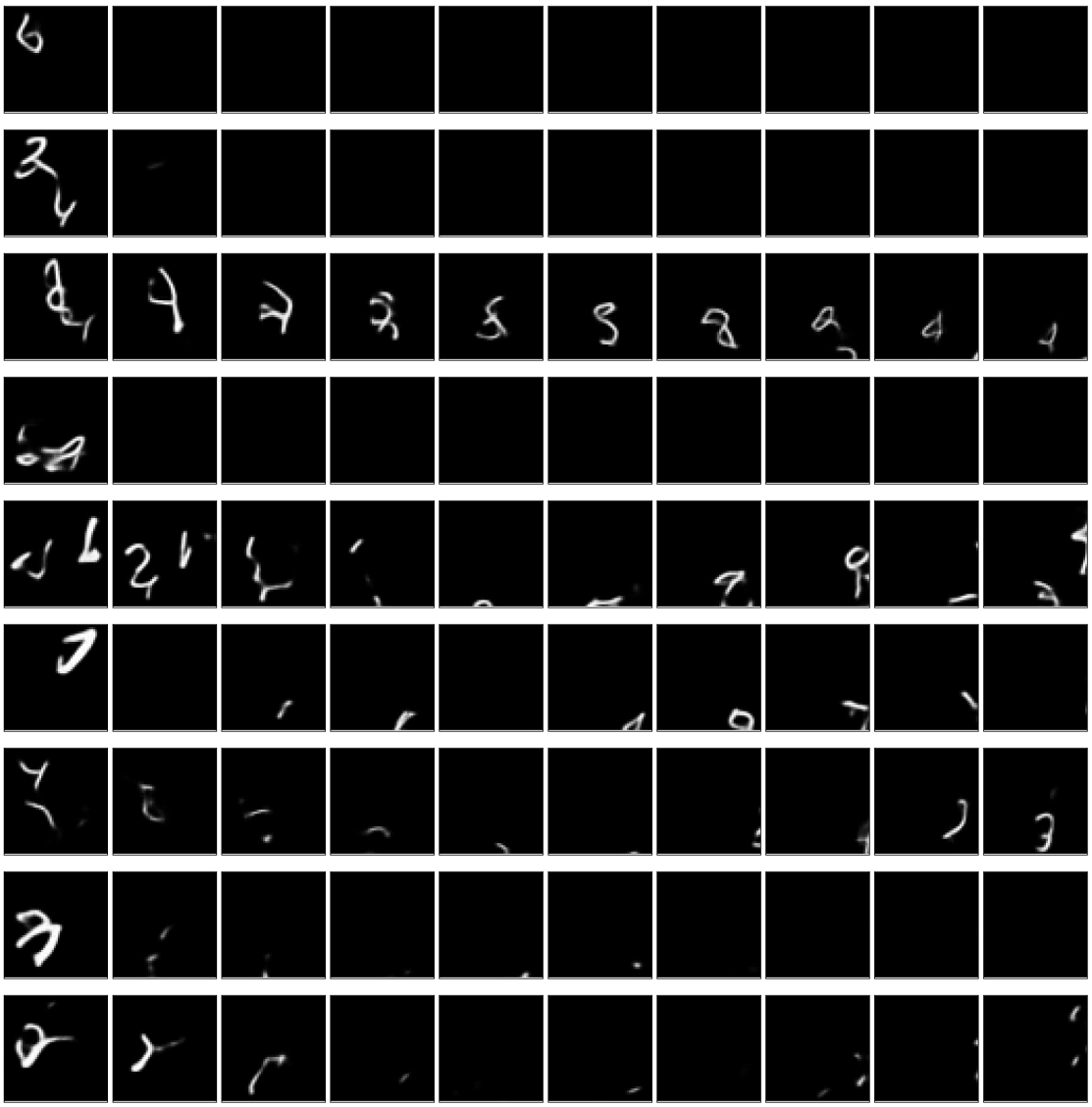


Figure 5.H.4: Samples from CONV-VRNN. They show lack of temporal consistency. Objects in the generated frames change between consecutive time-steps and they do not resamble digits from the training set.

5.H.3 Conditional Generation

Figure 5.H.5: Conditional generation from SQAIR, which sees only the first three frames in every case. Top is the input sequence (and the remaining ground-truth), while bottom is reconstruction (first three time-steps) and then generation.

5.I Reconstruction and Samples from the DukeMTMC Dataset

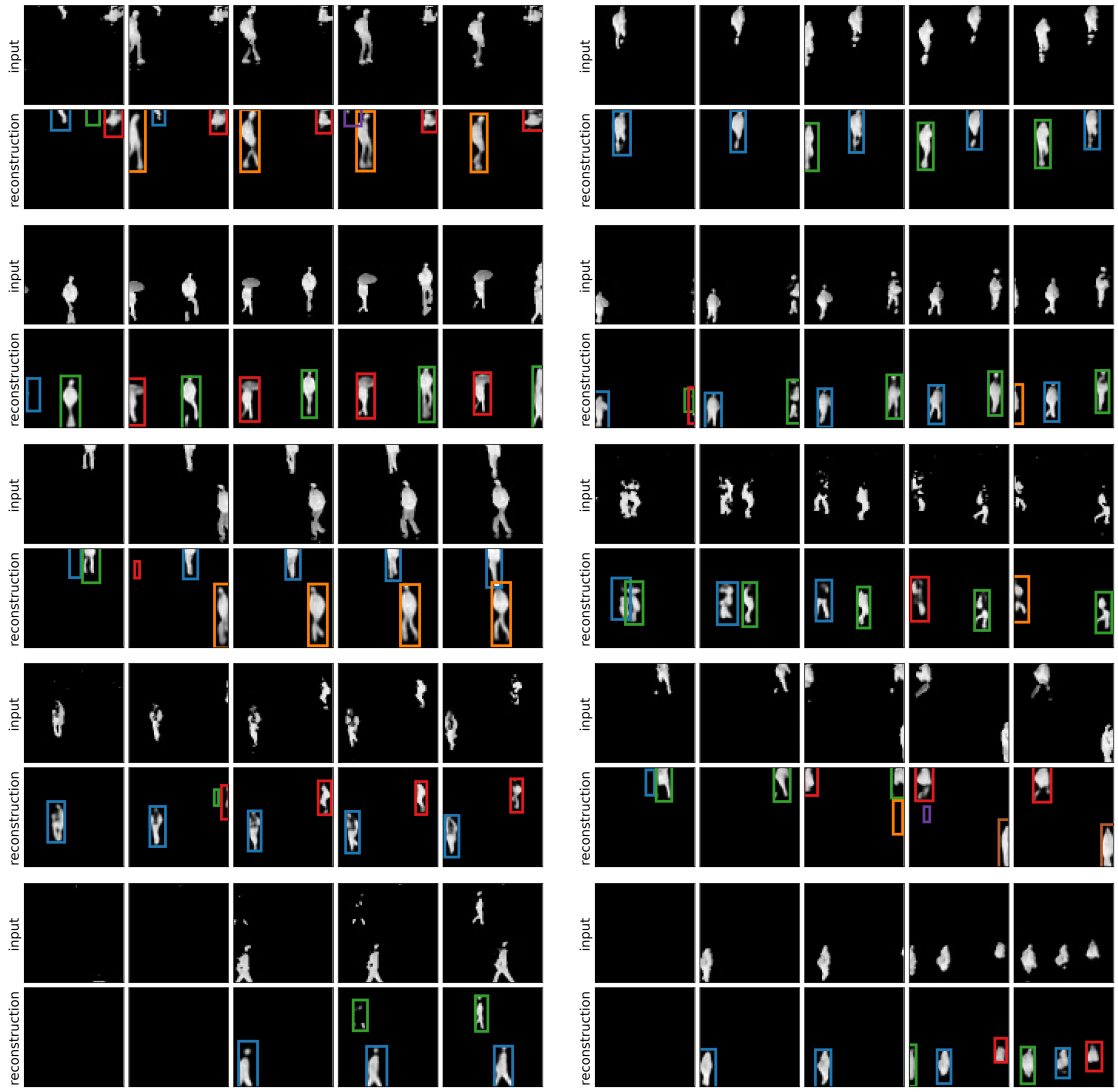


Figure 5.I.1: Sequences of input (first row) and SQAIR reconstructions with marked glimpse locations. While not perfect (spurious detections, missed objects), they are temporally consistent and similar in appearance to the inputs.

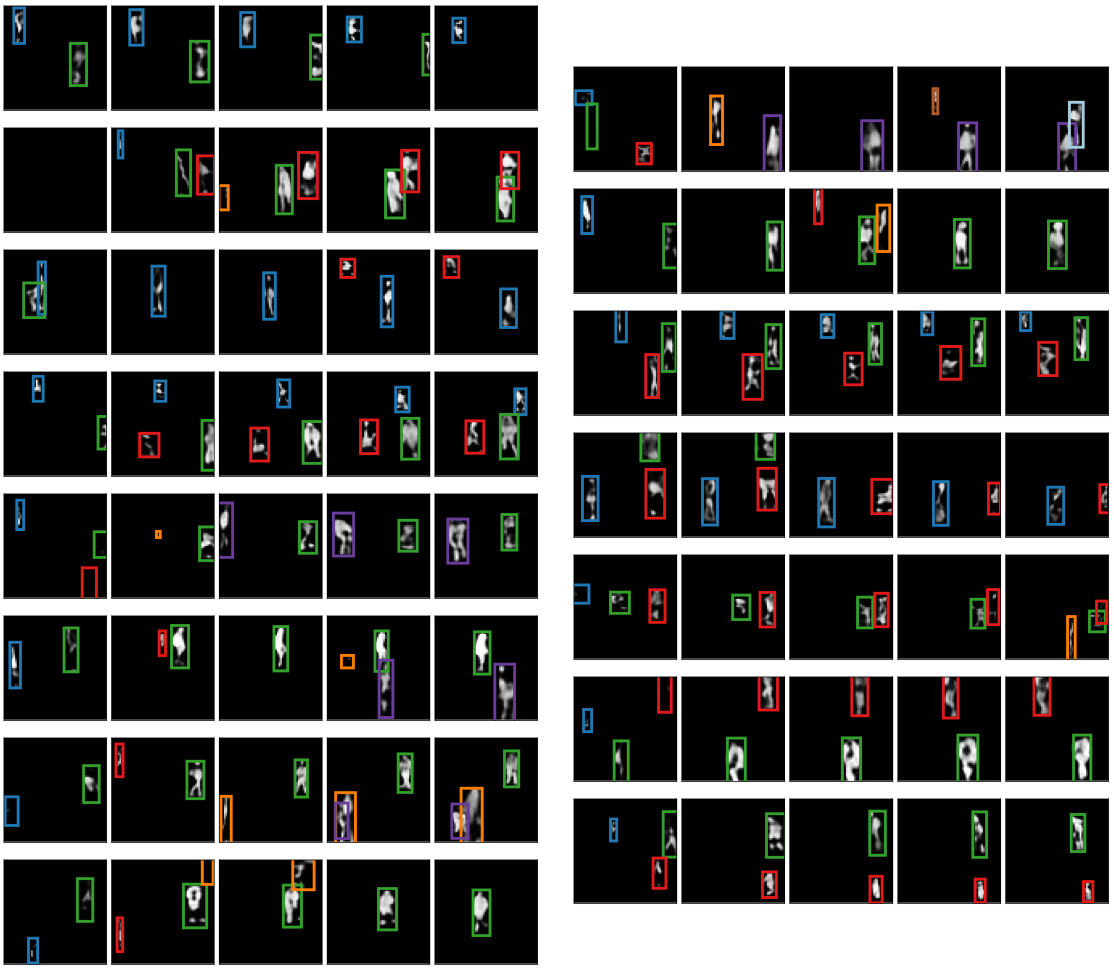


Figure 5.I.2: Samples with marked glimpse locations from SQAIR trained on the DukeMTMC dataset. Both appearance and motion is spatially consistent. Generated objects are similar in appearance to pedestrians in the training data. Samples are noisy, but so is the dataset.

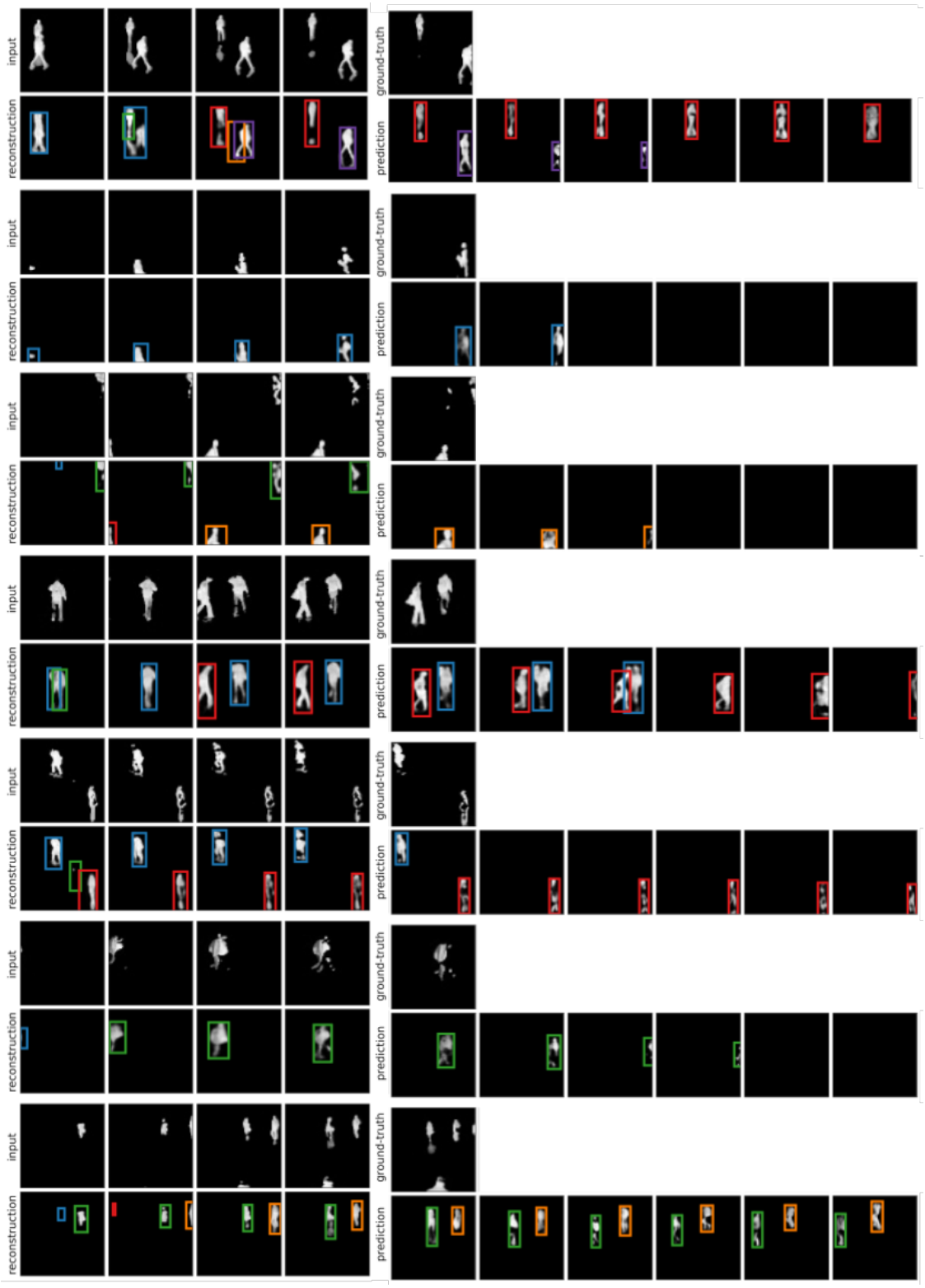


Figure 5.I.3: Conditional generation from SQAIR, which sees only the first four frames in every case. Top is the input sequence (and the remaining ground-truth), while bottom is reconstruction (first four time-steps) and then generation.

6

Stacked Capsule Autoencoders

Abstract

An object can be seen as a geometrically organized set of interrelated parts. A system that makes explicit use of these geometric relationships to recognize objects should be naturally robust to changes in viewpoint, because the intrinsic geometric relationships are viewpoint-invariant. We describe an unsupervised version of capsule networks, in which a neural encoder, which looks at all of the parts, is used to infer the presence and poses of object capsules. The encoder is trained by backpropagating through a decoder, which predicts the pose of each already discovered part using a mixture of pose predictions. The parts are discovered directly from an image, in a similar manner, by using a neural encoder, which infers parts and their affine transformations. The corresponding decoder models each image pixel as a mixture of predictions made by affine-transformed parts. We learn object- and their part-capsules on unlabeled data, and then cluster the vectors of presences of object capsules. When told the names of these clusters, we achieve state-of-the-art results for unsupervised classification on SVHN (55%) and near state-of-the-art on MNIST (98.5%).

6.1 Introduction

CNN work better than networks without weight-sharing because of their inductive bias: if a local feature is useful in one image location, the same feature is likely to be useful in other locations. It is tempting to exploit other effects of viewpoint changes by replicating features across scale, orientation and other affine degrees of freedom, but this quickly leads to cumbersome high-dimensional feature maps.

An alternative to replicating features across the non-translational degrees of freedom is to explicitly learn transformations between the natural coordinate frame of a whole object and the natural coordinate frames of each of its parts. Computer graphics relies on such object→part coordinate transformations to represent the geometry of an object in a viewpoint-invariant manner. Moreover, there is strong evidence that, unlike standard CNNs, human vision also relies on coordinate frames: imposing an unfamiliar coordinate frame on a familiar object makes it difficult to recognize the object or its geometry (**Rock73**; **Hinton79**).

A neural system can learn to reason about transformation between objects, their parts and the viewer, but each of the transformations is likely to require different representation. An object-part-relationship (OP) is viewpoint-invariant and is naturally coded by learned weights. The relationship of an object or part to the viewer changes with the viewpoint (it is viewpoint-equivariant) and is naturally coded using neural activations[1]. With this representation, pose of a single object is represented by its relationship to the viewer. Consequently, representing a single object does not necessitate replicating neural activations across space, unlike in CNNs. It is only processing two (or more) different instances of the same type of object in parallel that requires spatial replicas of both model parameters and neural activations.

In this paper we propose the SCAE, which has two stages (Fig. 4.1.1). The first stage, the Part Capsule Autoencoder (PCAE), segments an image into constituent

[1]This may explain why accessing perceptual knowledge about objects, when they are not visible, requires creating a mental image of the object with a specific viewpoint.

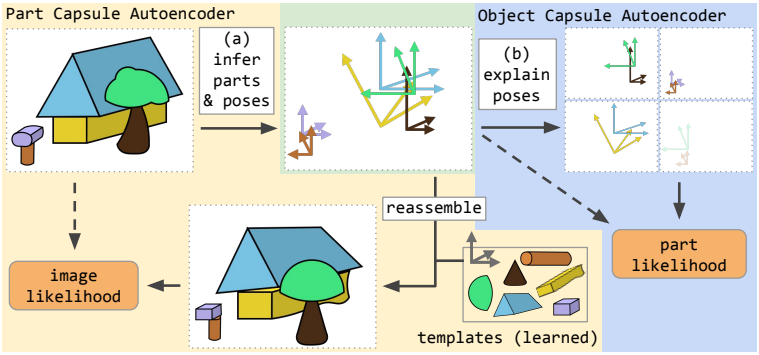


Figure 6.1.1: Stacked Capsule Autoencoder (SCAE): (a) *part* capsules segment the input into parts and their poses. The poses are then used to reconstruct the input by affine-transforming learned templates. (b) *object* capsules try to arrange inferred poses into objects, thereby discovering underlying structure. SCAE is trained by maximizing image and part log-likelihoods subject to sparsity constraints.

parts, infers their poses, and reconstructs each image pixel as a mixture of the pixels of transformed part templates. The second stage, the Object Capsule Autoencoder (OCAE), tries to organize discovered parts and their poses into a smaller set of objects that can explain the part poses using a separate mixture of predictions for each part. Every object capsule contributes components to each of these mixtures by multiplying its pose—the object-viewer-relationship (OV)—by the relevant object-part-relationship (OP)[2].

Stacked Capsule Autoencoders (Section 4.2) capture spatial relationships between whole objects and their parts when trained on unlabelled data. The vectors of presence probabilities for the object capsules tend to form tight clusters, and when we assign a class to each cluster we achieve state-of-the-art results for unsupervised classification on SVHN (55%) and near state-of-the-art on MNIST (98.5%), which can be further improved to 67% and 99%, respectively, by learning fewer than 300 parameters. We also present promising proof-of-concept results on CIFAR10 (Section 6.4). We describe related work in Section 4.4 and discuss implications of our work and future directions in Section 6.5.

[2]The type of a part capsule may determine which, if any, of an object's parts contribute to the mixture used to model the pose of an already discovered part

6.2 Stacked Capsule Autoencoders (scae)

Segmenting an image into parts is non-trivial, so we begin by abstracting away pixels and the part-discovery stage, and develop the CCAE (Section 4.2.1). It uses two-dimensional points as parts, and their coordinates are given as the input to the system. CCAE learns to model sets of points as arrangements of familiar constellations, each of which has been transformed by an independent similarity transform. The CCAE learns to assign individual points to their respective constellations—without knowing the number of constellations or their individual shapes in advance. Next, in Section 2.2, we develop the Part Capsule Autoencoder (PCAE) which learns to infer parts and their poses from images. Finally, we stack the Object Capsule Autoencoder (OCAE), which closely resembles the CCAE, on top of the PCAE to form the Stacked Capsule Autoencoder (SCAE).

6.2.1 Constellation Autoencoder (ccae)

Let $\{\mathbf{x}_m \mid m = 1, \ldots, M\}$ be a set of two-dimensional input points, where every point belongs to a constellation as in Figure 4.2.1. We first encode all input points (which take the role of part capsules) with Set Transformer (**Lee2019set**)—a permutation-invariant encoder h^{caps} based on attention mechanisms—into K object capsules. An object capsule k consists of a capsule feature vector $\mathbf{c}_k$, its presence probability $a_k \in [0, 1]$ and a 3×3 object-viewer-relationship (OV) matrix, which represents the affine transformation between the object (constellation) and the viewer. Note that each object capsule can represent only one object at a time. Every object capsule uses a separate MLP $\text{h}_\text{k}^{\text{part}}$ to predict $N \leq M$ part candidates from the capsule feature vector $\mathbf{c}_k$. Each candidate consists of the conditional probability $a_{k,n} \in [0, 1]$ that a given candidate part exists, an associated scalar standard deviation $\lambda_{k,n}$, and a 3×3 object-part-relationship (OP) matrix, which represents the affine transformation between the object capsule and the candidate

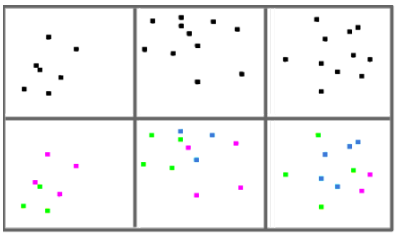


Figure 6.2.1: Unsupervised segmentation of points belonging to up to three constellations of squares and triangles at different positions, scales and orientations. The model is trained to reconstruct the points (top row) under the CCAE mixture model. The bottom row colors the points based on the parent with highest posterior probability in the mixture model. The right-most column shows a failure case. Note that the model uses sets of points, not pixels, as its input; we use images only to visualize the constellation arrangements.

part[3]. Candidate predictions $\mu_{k,n}$ are given by the product of the object capsule OV and the candidate OP matrices. We then model each input part as a Gaussian mixture, where $\mu_{k,n}$ and $\lambda_{k,n}$ are the centers and standard deviations of the isotropic components. See Figures 4.1.1 and 4.2.4 for illustration; formal description follows:

$$\mathrm{OV}_{1:K}, \mathbf{c}_{1:K}, a_{1:K} = \mathrm{h}^{\mathrm{caps}}(\mathbf{x}_{1:M}) \qquad \text{encode object capsule parameters,} \tag{6.1}$$

$$\mathrm{OP}_{k,1:N}, a_{k,1:N}, \lambda_{k,1:N} = \mathrm{h}_{\mathrm{k}}^{\mathrm{part}}(\mathbf{c}_k) \qquad \text{decode candidate parameters from } c_k\text{'s,} \tag{6.2}$$

$$V_{k,n} = \mathrm{OV}_k \mathrm{OP}_{k,n} \qquad \text{decode a part pose candidate,} \tag{6.3}$$

$$p(\mathbf{x}_m \mid k, n) = \mathcal{N}(\mathbf{x}_m \mid \mu_{k,n}, \lambda_{k,n}) \qquad \text{turn candidates into mixture components,} \tag{6.4}$$

$$p(\mathbf{x}_{1:M}) = \prod_{m=1}^{M} \sum_{k=1}^{K} \sum_{n=1}^{N} \frac{a_k a_{k,n}}{\sum_i a_i \sum_j a_{i,j}} p(\mathbf{x}_m \mid k, n)\,. \tag{6.5}$$

The model is trained without supervision by maximizing the likelihood of part capsules in Equation (4.5) subject to sparsity constraints, *cf.* Section 4.2.4. The part capsule m can be assigned to the object capsule $k^\star$ as $k^\star = \arg\max_k \; a_k a_{k,n}\, p(\mathbf{x}_m \mid k, n)$.[4] Empirical results show that this model is able to perform unsupervised instance-level segmentation of points belonging to different constellations, even in data

[3]Deriving these matrices from the capsule feature vector allows for deformable objects. We model OPs as the sum of an input-dependent component and a constant bias. We encourage different capsules to specialize to different constellations by putting a strong L_2 penalty on the former.

[4]We treat parts as independent and evaluate their probability under the same mixture model. While there are no clear 1:1 connections between parts and predictions, it seems to work well in practice.

which is difficult to interpret for humans. See Figure 4.2.1 for an example and Section 4.3.1 for details.

6.2.2 Part Capsule Autoencoder (pcae)

Explaining images as geometrical arrangements of parts requires first inferring what parts the images are composed of, as well as the relationships of the parts to the viewer (which we call their poses). For the CCAE a part is just a 2D point, but here each part capsule has a six degree of freedom (DOF) pose, a presence variable and a unique identity. We frame the part-discovery problem as auto-encoding: the encoder learns to infer the poses and presences of different part capsules, while the decoder learns an image template for each part (Fig. 4.2.2) similar to **Tieleman2014thesis**; **Eslami2016**. The templates corresponding to present parts are affine-transformed using their poses, and the pixels of these transformed templates are used to create a separate mixture model for each image pixel. The PCAE is followed by an Object Capsule Autoencoder (OCAE), which closely resambles the CCAE and is described in Section 4.2.3.

Let $\mathbf{y} \in [0,1]^{h \times w \times c}$ be the image. We limit the maximum number of part capsules to M and use an encoder to infer their poses $\mathbf{x}_m \in \mathbb{R}^6$, presence probabilities $d_m \in [0,1]$, and special features $\mathbf{z}_m \in \mathbb{R}^{c_z}$, one per part capsule. The latter do not take part in direct image reconstruction, but inform the OCAE about special aspects of the corresponding part; they are trained by backpropagating derivatives from the OCAE.

At present, we do not allow multiple occurrences of the same type of part in an image, so the part capsules themselves are not replicated across space, though they could be. However, we do need to recognize the part wherever it occurs in the image, and therefore the encoder consists of a CNN with a bottom-up attention mechanism; for every part capsule k, it predicts a feature map $\mathbf{e_k}$ of $6\,(\text{pose}) + 1\,(\text{presence}) + c_z\,(\text{special features})$ capsule parameters with spatial dimensions $h_e \times w_e$, as well as a single-channel attention mask $\mathbf{a_k}$. The final parameters for that capsule are computed as $\sum_i \sum_j \mathbf{e_{k,i,j}}\,\text{softmax}(\mathbf{a})_{\mathbf{k,i,j}}$, where softmax is along the spatial dimensions. This is similar to global average pooling,

but allows some spatial locations to contribute to the final result more than others; we call this approach *attention-based pooling*. Its effect on the model performance is analyzed in Section 4.3.3.

The image pixels are modelled as independent Gaussian mixtures. For every pixel, we take the corresponding pixels of the transformed templates and treat them as centers of isotropic Gaussian components with constant variance. Their mixing probabilities are proportional to both presence probabilities of part capsules and a function $f_c : \mathbb{R}^c \mapsto [0, 1]$ of the color value at that location[5], where c is the number of image channels. More formally:

$$\mathbf{x}_{1:M}, d_{1:M}, \mathbf{z}_{1:M} = \mathrm{h}^{\mathrm{enc}}(\mathbf{y}) \qquad \text{encode the image to part capsule parameters,} \tag{6.6}$$

$$\widehat{T}_m = \mathrm{TransformImage}(T_m, \mathbf{x}_m) \qquad \text{apply affine transforms to image templates,} \tag{6.7}$$

$$p^y_{m,i,j} \propto d_m \, \mathrm{f_c}\left(\widehat{T}_{m,i,j}\right) \qquad \text{compute mixing probabilities,} \tag{6.8}$$

$$p(\mathbf{y}) = \prod_{i,j} \sum_{m=1}^{M} p^y_{m,i,j} \, \mathcal{N}\left(y_{i,j} \mid \widehat{T}_{m,i,j}, \sigma_y^2\right) \qquad \text{calculate image likelihood.} \tag{6.9}$$

6.2.3 Object Capsule Autoencoder (ocae)

The next step is to find objects in the already discovered parts[6]. To do so, we use concatenated poses $\mathbf{x}_m$, special features $\mathbf{z}_m$ and flattened templates T_m (which convey the identity of the part capsule) as an input to the OCAE, which differs from the CCAE in the following ways. Firstly, we feed part capsule presence probabilities d_m into the OCAE's encoder—these are used to bias the Set Transformer's attention mechanism to not take absent points into account. Secondly, d_m's are also used to weigh the part-capsules' log-likelihood, *cf.* Equation (4.5). Additionally, we stop the gradient on all of OCAE's inputs except the special features to improve training stability

[5]Templates are assumed to be sparse; if there exists a template that has a non-zero value at a given location, then this templates should be used.

[6]Discovered objects are *not* used top-down to refine the presences or poses of the parts during inference. However, the derivatives backpropagated via OCAE refine the lower-level encoder network that infers the parts.

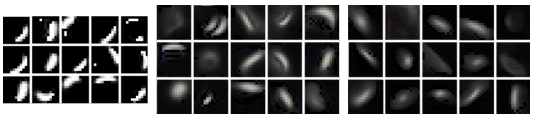


Figure 6.2.2: Templates learned on MNIST (left) as well as sobel-filtered SVHN (middle) and CIFAR10 (right). In each case templates converge to strokes. For SVHN they often take the form of double strokes—this is due to sobel filtering, which effectively extracts edges.

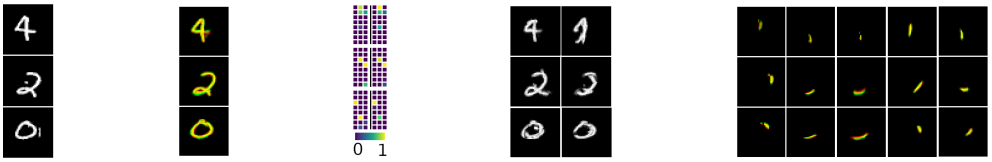


Figure 6.2.3: 40×40 MNIST (a) images and their (b) reconstructions from part capsules in red and object capsules in green, with overlapping regions in yellow. Only a few object capsules are activated for every input (c) a priori (left) and even fewer are needed to reconstruct it (right). The most active capsules (d) capture object identity and the majority of information about its appearance. Finally, (e) affine-transformed templates show how exactly parts are used to reconstruct the images.

and avoid the problem of collapsing latent variables; see e. g., **Rasmus2015ladder**. Finally, parts discovered by the PCAE have independent identities (templates and special features rather than 2D points). Therefore, every part-pose is explained as an independent mixture of predictions from object-capsules—where every object capsule makes exactly M candidate predictions $V_{k,1:M}$, or exactly **one** candidate prediction per part. Consequently, the part-capsule likelihood is given by,

$$p(\mathbf{x}_{1:M}, d_{1:M}) = \prod_{m=1}^{M} \left[\sum_{k=1}^{K} \frac{a_k a_{k,m}}{\sum_i a_i \sum_j a_{i,j}} p(\mathbf{x}_m \mid k, m) \right]^{d_m} . \tag{6.10}$$

6.2.4 Achieving Sparse and Diverse Capsule Presences

Stacked Capsule Autoencoders are trained to maximise pixel and part log-likelihoods ($\mathcal{L}_{\text{ll}} = \log p(\mathbf{y}) + \log p(\mathbf{x}_{1:M})$). If not constrained, however, they tend to either use all of the part and object capsules to explain every data example, or collapse onto using always the same subset of capsules, regardless of the input. We would like the model to use different sets of part-capsules for different input examples and to specialize object-capsules to particular arrangements of parts; to encourage this, we impose sparsity and entropy constraints. We evaluate their importance in Section 4.3.3.

We first define prior and posterior object-capsule presence as follows. For a minibatch of size B with K object capsules and M part capsules we define a minibatch of prior capsule presence $a^{\text{prior}}_{1:K}$ with dimension $[B, K]$ and posterior capsule presence $a^{\text{posterior}}_{1:K,1:M}$ with dimension $[B, K, M]$ as,

$$a^{\text{prior}}_{k} = a_k \max_m a_{m,k}\,, \qquad a^{\text{posterior}}_{k,m} = a_k a_{k,m}\,\mathcal{N}(\mathbf{x}_m \mid m, k)\,, \tag{6.11}$$

respectively; the former is the maximum presence probability among predictions from object capsule k while the latter is the unnormalized mixing probability used to explain part capsule m.

Prior sparsity Let $\overline{u}_k = \frac{1}{B}\sum_{b=1}^{B} a^{\text{prior}}_{b,k}$ the average presence probability of the object capsule k among different training examples, and $\widehat{u}_b = \sum_{k=1}^{K} a^{\text{prior}}_{b,k}$ the sum of object capsule presence probabilities for a given example. If we assume that training examples contain objects from different classes uniformly at random and we would like to assign the same number of object capsules to every class then each class would obtain K/C capsules. Moreover, if we assume that only one object is present in every image, then B/C object capsules should be present for every input example. To this end, we minimize,

$$\mathcal{L}_{\text{prior}} = \frac{1}{B}\sum_{b=1}^{B} ||\widehat{u}_b - \frac{K}{C}||_2 + \frac{1}{K}\sum_{k=1}^{K} ||\overline{u}_k - \frac{B}{C}||_2\,. \tag{6.12}$$

Posterior Sparsity Similarity, let $\overline{v}_k$ and $\widehat{v}_b$ be the the normalized versions of $\sum_{k,m} a^{\text{posterior}}_{b,k,m}$ and $\sum_{b,m} a^{\text{posterior}}_{b,k,m}$, respectively. We find it beneficial to minimize the within-example entropy of capsule posterior presence $\mathcal{H}(\overline{v}_k)$ and maximize its between-example entropy $\mathcal{H}(\widehat{v}_b)$, where $\mathcal{H}$ is the entropy . The final loss reads as,

$$\mathcal{L}_{\text{posterior}} = \frac{1}{K}\sum_{k=1}^{K} \mathcal{H}(\overline{v}_k) - \frac{1}{B}\sum_{b=1}^{B} \mathcal{H}(\widehat{v}_b)\,. \tag{6.13}$$

ak: say why it is beneficial

Every active object capsule should explain at least two parts We say that an object capsule has 'won' a part if it has the highest posterior mixing probability for that part among other object capsules. We then create binary labels for each of object capsules, where the label is 1 if the capsule wins at least two parts and it is 0 otherwise. The final loss takes the form of binary cross-entropy between the generated label and the prior capsule presence. This loss is used only for the

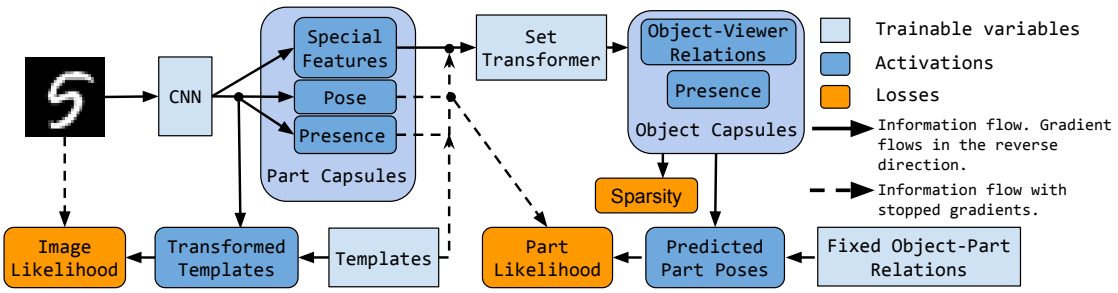


Figure 6.2.4: SCAE architecture.

stand-alone constellation model experiments on point data, *cf.* Sections 4.2.1 and 4.3.1.

Fig. 4.2.4 shows the schematic architecture of SCAE. We optimize a weighted sum of image and part likelihoods and the auxiliary losses. Loss weight selection process as well as the values used for experiments are explained in Section 4.A.

In order to make the values of presence probabilities ($a_k, a_{k,m}$ and d_m) closer to binary we inject uniform noise $\in [-2, 2]$ into logits, similar to **Tieleman2014thesis**. This forces the model to predict logits that are far from zero to avoid stochasticity and makes the predicted presence probabilities close to binary. Interestingly, it tends to work better in our case than using the Concrete distribution (**Maddison2017concrete**).

6.3 Evaluation

The decoders in the SCAE use explicitly parameterised affine transformations that allow the encoders' inputs to be explained with a small set of transformed objects or parts. The following evaluations show how the embedded geometrical knowledge helps to discover patterns in data. Firstly, we show that the CCAE discovers underlying structures in arrangements of constellations made of two-dimensional points, thereby performing instance-level segmentation. Secondly, we pair an OCAE with a PCAE and investigate whether the resulting SCAE can discover structure in real images. Finally, we present an ablation study that shows which components of the model contribute to the results.

6.3.1 Discovering Constellations

We create arrangements of constellations online, where every input example consists of up to 11 two-dimensional points belonging to up to three different constellations (two squares and a triangle) as well as binary variables indicating presence of the points (points can be missing). Each constellation is included with probability 0.5 and undergoes a similarity transformation, whereby it is randomly scaled, rotated by up to 180° and shifted. Finally, every input example is normalized such that all points lie within $[-1, 1]^2$. Note that we use sets of points, and not images, as inputs to our model.

We compare the CCAE against a baseline that uses the same encoder but a simpler decoder: the decoder uses the capsule parameter vector $\mathbf{c}_k$ to directly predict the location, precision and presence probability of each of the four points as well as the presence probability of the whole corresponding constellation. Implementation details are listed in Section 4.A.1.

Both models are trained unsupervised by maximizing the part log-likelihood. We evaluate them by trying to assign each input point to one of the object capsules. To do so, we assign every input point to the object capsule with the highest posterior probability for this point, *cf.* Section 4.2.1, and compute segmentation accuracy (i. e., the true-positive rate).

The CCAE consistently achieves below 4% error with the best model achieving 2.8%, while the best baseline achieved 26% error using the same budget for hyperparameter search. This shows that wiring in an inductive bias towards modelling geometric relationships can help to bring down the error by an order of magnitude—at least in a toy setup where each set of points is composed of familiar constellations that have been independently transformed.

6.3.2 Unsupervised Class Discovery

To allow for multimodality in the appearance of objects of a specific class, we typically use more object capsules than the number of class labels. We expect that the vector of presence probabilities of object capsules should be highly informative of

Method	MNIST	CIFAR10	SVHN
KMEANS (**adc**)	53.49	20.8	12.5
AE (**ae**)§	81.2	31.4	-
GAN (**gan**)§	82.8	31.5	-
IMSAT (**imsat**)†,∇	**98.4** (0.4)	45.6 (0.8)	**57.3** (3.9)
IIC (**iic**)§,†	**98.4** (0.6)	**57.6** (5.0)	-
ADC (**adc**)†	**98.7** (0.6)	29.3 (1.5)	38.6 (4.1)
MAX-ACT (SCAE)	**98.0 (.15)**	19.79 (1.0)	49.07 (1.7)
CLUST-NN (SCAE)	**98.5 (.11)**	19.39 (1.5)	**53.0 (3.8)**
LIN-MATCH (SCAE)	**98.5 (.10)**	25.01 (1.0)	**55.33 (3.4)**
LIN-PRED (SCAE)	**98.9 (.07)**	33.48 (0.3)	**67.27 (4.5)**

Table 6.3.1: Unsupervised classification results in % with (standard deviation) are averaged over 5 runs. Methods based on mutual information are shaded. Results marked with † use data augmentation, ∇ use IMAGENET-pretrained features instead of images, while § are taken from **iic**. We highlight the best results and those that are are within its 98% confidence interval according to a two-sided t test.

the class label. To test this hypothesis, we train SCAE on MNIST, SVHN and CIFAR10 and try to assign class labels to vectors of object capsule presences. This is done with one of the following methods: MAX-ACT: we search for a training example that maximally activates given object capsule and assign the corresponding label to this capsule; CLUSTER-NN: we perform KMEANS clustering into C clusters and then find the training example that is the closest to each cluster's centroid to assign a label to the cluster; LIN-MATCH: after finding 10 clusters[7] with KMEANS we use bipartite graph matching (**Kuhn1955hungarian**) to find the permutation of cluster indices that minimizes the classification error—this is standard practice in unsupervised classification, see e. g., **iic**; LIN-PRED: we train a linear classifier with supervision given the presence vectors; this learns $K \times 10$ weights and 10 biases, where K is the number of object capsules, but it does not modify any parameters of the main model.

In agreement with previous work on unsupervised clustering (**iic**; **imsat**; **Hjelm2019deepinfomax**; **adc**), we train our models and report results on full datasets (TRAIN, VALID and TEST splits). The linear transformation used in LIN-PRED variant of our method is trained on the TRAIN split of respective datasets while its performance on the TEST split is reported.

[7] All considered datasets have 10 classes.

We used an PCAE with 24 single-channel 11×11 templates for MNIST and 24 and 32 three-channel 14×14 templates for SVHN and CIFAR10, respectively. We used sobel-filtered images as the reconstruction target for SVHN and CIFAR10, as in **jaiswal**, while using the raw pixel intensities as the input to PCAE. The OCAE used 24, 32 and 64 object capsules, respectively. Further details on model architectures and hyper-parameter tuning are available in Section 4.A. All results are presented in Table 4.3.1. SCAE achieves competitive results in unsupervised object classification on MNIST and SVHN and under-performs slightly on CIFAR10, which is further discussed in Section 6.5.

6.3.3 Ablation study

SCAEs have many moving parts; an ablation study shows which model components are important and to what degree. We train SCAE variants on MNIST as well as a padded-and-translated 40×40 version of the dataset, where the original digits are translated up to 6 pixels in each direction. Trained models are tested on TEST splits of both datasets; additionally, we evaluate the model trained on the 40×40 MNIST on the TEST split of AFFNIST dataset. Testing on AFFNIST shows whether the model can generalize to unseen viewpoints. This task was used by **sparsecaps** to evaluate Sparse Unsupervised Capsules, which achieved 90.12% accuracy. SCAE achieves $92.2 \pm 0.59\%$, which indicates that it is better at viewpoint generalization. We choose the LIN-MATCH performance metric, since it is the one favoured by the unsupervised classification community. Results are split into several groups and shown in Table 4.3.2. We describe each group in turn. Group a) shows that sparsity losses introduced in Section 4.2.4 increase model performance, but that the posterior loss might not be necessary. Group b) checks the influence of injecting noise into logits for presence probabilities, *cf.* Section 4.2.4. Injecting noise into part capsules seems critical, while noise in object capsules seems unnecessary—the latter might be due to sparsity losses. Group c) shows that using similarity (as opposed to affine) transforms in the decoder can be restrictive in some cases, while not allowing deformations hurts performance in every case.

	Method	MNIST	40 × 40 MNIST	AFFNIST
	full model	**97.0 (.87)**	**98.5 (.1)**	**92.2 (.59)**
a)	no posterior sparsity	**96.7 (.7)**	**98.2 (.48)**	87.6 (1.63)
	no prior sparsity	90.5 (7.56)	94.0 (3.03)	74.0 (4.94)
	no prior/posterior sparsity	63.0 (13.48)	62.7 (10.46)	40.7 (6.81)
b)	no noise in object caps	**96.4 (1.41)**	**97.8 (.67)**	**90.8 (2.97)**
	no noise in any caps	84.8 (6.22)	85.1 (13.13)	76.3 (12.89)
	no noise in part caps	83.9 (7.57)	80.2 (9.1)	73 (9.04)
c)	similarity transforms	90.4 (13.78)	**97.4 (.99)**	**90.1 (2.62)**
	no deformations	87.6 (6.13)	95.2 (1.04)	87.6 (1.26)
d)	LINEAR part enc	**94.8 (3.0)**	98.1 (.26)	76.3 (2.22)
	CONV part enc	**96.3 (.85)**	**97.8 (.95)**	80.1 (2.58)
e)	MLP enc for object caps	73.0 (6.34)	70.3 (11.2)	52.5 (11.29)
f)	no special features	63.1 (10.55)	66.9 (23.59)	50.5 (18.26)

Table 6.3.2: Ablation study on MNIST. All used model components contribute to its final performance. AFFNIST results show out-of-distribution generalization properties and come from a model trained on 40 × 40 MNIST. Numbers represent average % and (standard deviation) over 10 runs. We highlight the best results and those that are are within its 98% confidence interval according to a two-sided t test.

Group d) evaluates the type of the part-capsule encoder. The LINEAR encoder entails a CNN followed by a fully-connected layer, while the CONV encoder predicts one feature map for every capsule parameter, followed by global-average pooling. The choice of part-capsule encoder seems not to matter much for within-distribution performance; however, our attention-based pooling does achieve much higher classification accuracy when evaluated on a different dataset, showing better generalization to novel viewpoints.

Additionally, e) using Set Transformer as the object-capsule encoder is essential. We hypothesize that it is due to the natural tendency of Set Transformer to find clusters, as reported in **Lee2019set**. Finally, f) using special features $\mathbf{z}_m$ seems not less important—presumably due to effects the high-level capsules have on the representation learned by the primary encoder.

6.4 Related Work

Capsule Networks Our work combines ideas from Transforming Autoencoders (**Hinton2011tae**) and EM Capsules (**Hinton2018capsule**). Transforming au-

toencoders discover affine-aware capsule *instantiation parameters* by training an autoencoder to predict an affine-transformed version of the input image from the original image plus an extra input, which explicitly represents the transformation. By contrast, our model does not need any input other than the image.

Both EM Capsules and the preceding Dynamic Capsules (**Sabour2017capsule**) use the poses of parts and learned part→object relationships to vote for the poses of objects. When multiple parts cast very similar votes, the object is assumed to be present, which is facilitated by an interactive inference (routing) algorithm. Iterative routing is inefficient and has prompted further research. **wang2018optimization** formulated routing as an optimization of a clustering loss and a KL-divergence-based regularization term. **zhang2018fast** proposed a weighted kernel density estimation-based routing method. **encapsule** proposed approximating routing with two branches and sending feedback via optimal transport divergence between two distributions (lower and higher capsules). In contrast to prior work, we use objects to predicts parts rather than vice-versa, therefore we can dispense with iterative routing at inference time. The encoder of the OCAE learns how to group parts into objects and it respects the single parent constraint, because it is trained using derivatives produced by a decoder that uses a mixture model of parts which assumes that each part must be explained by a single object.

Additionally, since it is the objects that predict parts, the parts are allowed to have fewer degrees-of-freedom in their poses than objects (as in the CCAE). Inference is still possible, because the OCAE encoder makes object predictions based on *all* the parts rather than an individual part.

A further advantage of our version of capsules is that it can perform unsupervised learning. Previous versions of capsules used discriminative learning, though **sparsecaps** used the reconstruction MLP introduced in **Sabour2017capsule** to train Dynamic Capsules without supervision and has shown that unsupervised training for capsule-conditioned reconstruction helps with generalization to AFFNIST classification; we further improve on their results, *cf.* Section 4.3.3.

Unsupervised Classification There are two main approaches to unsupervised object category detection in computer vision. The first one is based on representation learning and typically requires discovering clusters or learning a classifier on top of the learned representation. **Eslami2016**; **Kosiorek2018sqair** use an iterative procedure to infer a variable number of latent variables, one for every object in a scene, that are highly informative of object class, while **Greff2019multi**; **Burgess2019monet** perform unsupervised instance-level segmentation in an iterative fashion. While similar to our work, these approaches cannot decompose objects into their constituent parts and do not provide explicit description of object shape (e. g., templates and their poses in our model).

The second approach targets classification explicitly by minimizing mutual information (MI)-based losses and directly learning class-assignment probabilities. IIC (**iic**) maximizes an exact estimator of MI between two discrete probability vectors describing (transformed) versions of the input image. DeepInfoMax (**Hjelm2019deepinfomax**) relies on negative samples and maximizes MI between the predicted probability vector and its input via noise-contrastive estimation (**Gutmann2010nce**). This class of methods directly maximizes the amount of information contained in an assignment to discrete clusters and they hold state-of-the-art results on most unsupervised classification tasks. MI-based methods suffer from typical drawbacks of mutual information estimation: they require heavy data augmentation and large batch sizes. This is in contrast to our method, which achieves comparable performance with batch size no bigger than 128 and with no data augmentation.

Geometrical Reasoning Other attempts at incorporating geometrical knowledge into neural networks include exploiting equivariance properties of group transformations (**Cohen2016group**) or new types of convolutional filters (**mallat**; **kocvok**). Although they achieve significant parameter efficiency in handling rotations or reflections compared to standard CNNs, these methods cannot handle additional degrees of freedom of affine transformations—like scale. **lenssen** combined capsule networks with group convolutions to guarantee equivariance

and invariance in capsule networks. Spatial Transformers (ST; **Jaderberg2015**) apply affine transformations to the image sampling grid while steerable networks (**Cohen2016steerable**; **Jacobsen2017dynamic**) dynamically change convolutional filters. These methods are similar to ours in the sense that transformation parameters are predicted by a neural network, but differ in the sense that ST uses global transformations applied to the whole image while steerable networks use only local transformations. Our approach can use different global transformations for every object as well as local transformations for each of their parts.

6.5 Discussion

The main contribution of our work is a novel method for representation learning, in which highly structured decoder networks are used to train one encoder network that can segment an image into parts and their poses and another encoder network that can compose the parts into coherent wholes. Despite the fact that our training objective is not concerned with classification or clustering, SCAE is the only method that achieves competitive results in unsupervised object classification without relying on mutual information (MI). This is significant, since unlike our method, MI-based methods require sophisticated data augmentation. It may be possible to further improve results by using an MI-based loss to train SCAE, where the vector of capsule probabilities could take the role of discrete probability vectors in IIC (**iic**). SCAE under-performs on CIFAR10, which could be because of using fixed templates, which are not expressive enough to model real data. This might be fixed by building deeper hierarchies of capsule autoencoders (e. g., complicated scenes in computer graphics are modelled as deep trees of affine-transformed geometric primitives) as well as using input-dependent shape functions instead of fixed templates—both of which are promising directions for future work. It may also be possible to make a much better PCAE for learning the primary capsules by using a differentiable renderer in the generative model that reconstructs pixels from the primary capsules.

Finally, the SCAE could be the 'figure' component of a mixture model that also includes a versatile 'ground' component that can be used to account for everything

except the figure. A complex image could then be analyzed using sequential attention to perceive one figure at a time.

6.6 Acknowledgements

We would like to thank Sandy H. Huang for help with editing the manuscript and making Figure 4.1.1. Additionally, we would like to thank S. M. Ali Eslami and Danijar Hafner for helpful discussions throughout the project. We also thank Hyunjik Kim, Martin Engelcke, Emilien Dupont and Simon Kornblith for feedback on initial versions of the manuscript.q

Appendix

6.A Model Details

6.A.1 Constellation Experiments

The CCAE uses a four-layer Set Transformer as its encoder. Every layer has four attention heads, 128 hidden units per head, and is followed by layer norm (**Ba2016LayerN**). The encoder outputs three 32-dimensional vectors—one for each object capsule. The decoder uses a separate neural net for each object capsule to predict all parameters used to model its points: this includes four candidate part predictions per capsule for a total of 12 candidates. In this experiment, each object→part relationship OP is just a 2-D offset in the object's frame of reference (instead of a 3×3 matrix) and it is affine transformed by the corresponding OV matrix to predict the 2-D point.

6.A.2 Image Experiments

We use a convolutional encoder for part capsules and a set transformer encoder (**Lee2019set**) for object capsules. Decoding from object capsule to part capsules is done with MLPs, while the input image is reconstructed with affine-transformed learned templates. Details of the architectures we used are available in Table 4.A.1.

Table 6.A.1: Architecture details. S in the last column means that the entry is the same as for SVHN.

Dataset	Constellation	MNIST	SVHN	CIFAR10
num templates	N/A	24	24	32
template size	N/A	11×11	14×14	S
num capsules	3	24	32	64
part CNN	N/A	2x(128:2)-2x(128:1)	2x(128:1)-2x(128:2)	S
set transformer	4x(4-128)-32	3x(1-16)-256	3x(2-64)-128	S

We use ReLu nonlinearities except for presence probabilities, for which we use sigmoids. (128:2) for a CNN means 128 channels with a stride of two. All kernels

are 3×3. For set transformer (1-16)-256 means one attention head, 16 hidden units and 256 output units; it uses layer normalization (**Ba2016LayerN**) as in the original paper (**Lee2019set**) but no dropout. All experiments (apart from constellations) used 16 special features per part capsule.

For SVHN and CIFAR10, we use normalized sobel-filtered images as the target of the reconstruction to emphasize the shape importance. Figure 4.B.1 in Section 4.B shows examples of SVHN and CIFAR10 reconstruction. The filtering procedure is as follows: 1) apply sobel filtering, 2) subtract the median color, 3) take the absolute value of the image, 4) normalize for image values to be $\in [0, 1]$.

All models are trained with the RMSProp optimizer (**tieleman2012rms**) momentum = .9 and $\epsilon = (10 * \text{batch\_size})^{-2}$. Batch size is 64 for constellations and 128 for all other datasets. The learning rate was equal to 10^{-5} for MNIST and constellation experiments (without any decay), while we run a hyperparameter search for SVHN and CIFAR10: we searched learning rates in the range of $5 * 10^{-5}$ to $5*10^{-4}$ and exponential learning rate decay of 0.96 every 10^3 or $3*10^3$ weight updates. Learning rate of 10^{-4} was selected for both SVHN and CIFAR10, the decay steps was 10^3 for SVHN and $3 * 10^3$ for CIFAR10. The LIN-PRED accuracy on a validation set is used as a proxy to select the best hyperparameters—including weights on different losses, reported in Table 4.A.2. Models were trained for up to $3 * 10^5$ iterations on single Tesla V100 GPUs, which took 40 minutes for constellation experiments and less than a day for CIFAR10.

Table 6.A.2: Loss weights values. The *within* and *between* quantifiers in sparsity losses corresponds to different terms of Equations (4.12) and (4.13).

Dataset	Constellation	MNIST	SVHN	CIFAR10
part ll weight	1	1	2.56	2.075
image ll weight	N/A	1	1	1
prior within sparsity	1	1	0.22	0.17
prior between sparsity	1	1	0.1	0.1
posterior within sparsity	0	10	8.62	1.39
posterior between sparsity	0	10	0.26	7.32
too-few-active-capsules	10	0	0	0

6.B Reconstructions

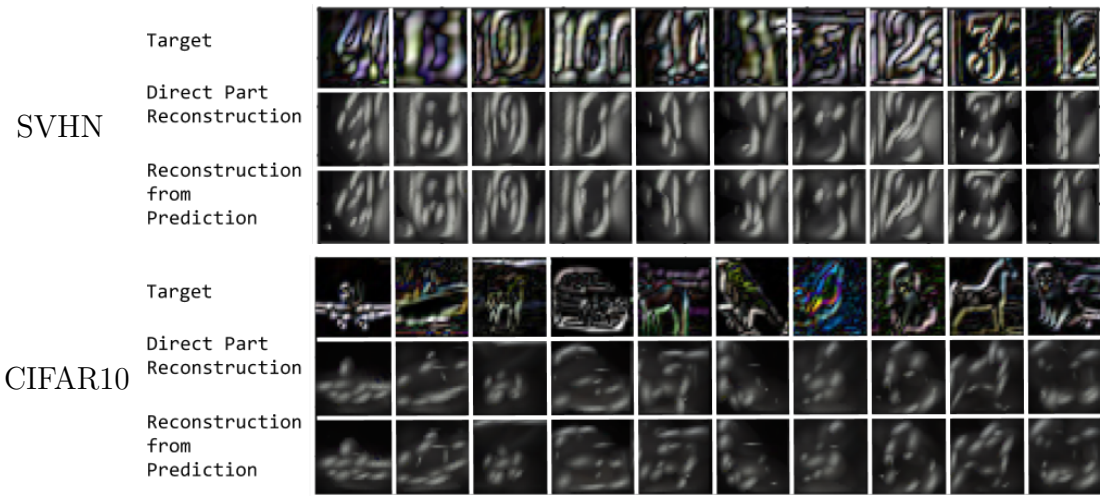


Figure 6.B.1: 10 Sample SVHN and Cifar10 reconstructions. First row shows Sobel filtered target image. Second row shows the reconstruction from Part Capsule Layer directly. Third row shows the reconstruction if we use the object predictions for the Part poses instead of Part poses themselves for reconstruction. The templates in this model has the same number of channels as the image, but they have converged to black and white templates and the reconstruction do not have color diversity. The SCAE model is trained completely unsupervised but the reconstructions tend to focus on the center digit in SVHN and filter the rest of the clutter.

7

Tighter Variational Bounds are Not Necessarily Better

Abstract

We provide theoretical and empirical evidence that using tighter ELBOs can be detrimental to the process of learning an inference network by reducing the signal-to-noise ratio of the gradient estimator. Our results call into question common implicit assumptions that tighter ELBOs are better variational objectives for simultaneous model learning and inference amortization schemes. Based on our insights, we introduce three new algorithms: the partially importance weighted auto-encoder (PIWAE), the multiply importance weighted auto-encoder (MIWAE), and the combination importance weighted auto-encoder (CIWAE), each of which includes the standard importance weighted auto-encoder (IWAE) as a special case. We show that each can deliver improvements over IWAE, even when performance is measured by the IWAE target itself. Furthermore, our results suggest that PIWAE may be able to deliver simultaneous improvements in the training of both the inference and generative networks.

7.1 Introduction

Variational bounds provide tractable and state-of-the-art objectives for training deep generative models (**kingma2014auto**; **rezende2014stochastic**). Typically taking the form of a lower bound on the intractable model evidence, they provide surrogate targets that are more amenable to optimization. In general, this optimization requires the generation of approximate posterior samples during the model training and so a number of methods simultaneously learn an *inference network* alongside the target *generative network*.

As well as assisting the training process, this inference network is often also of direct interest itself. For example, variational bounds are often used to train auto-encoders (**bourlard1988auto**; **hinton1994autoencoders**; **gregor2016towards**; **chen2016variational**), for which the inference network forms the encoder. Variational bounds are also used in amortized and traditional Bayesian inference contexts **hoffman2013stochastic**; **ranganath2014black**; **paige2016inference**; **le2017inference**, for which the generative model is fixed and the inference network is the primary target for the training.

The performance of variational approaches depends upon the choice of evidence lower bound (ELBO) and the formulation of the inference network, with the two often intricately linked to one another; if the inference network formulation is not sufficiently expressive, this can have a knock-on effect on the generative network (**burda2016importance**). In choosing the ELBO, it is often implicitly assumed that using tighter ELBOs is universally beneficial, at least whenever this does not in turn lead to higher variance gradient estimates.

In this work we question this implicit assumption by demonstrating that, although using a tighter ELBO is typically beneficial to gradient updates of the generative network, it can be detrimental to updates of the inference network. Remarkably, we find that it is possible to simultaneously tighten the bound, reduce the variance of the gradient updates, and *arbitrarily deteriorate* the training of the inference network.

Specifically, we present theoretical and empirical evidence that increasing the number of importance sampling particles, K, to tighten the bound in the IWAE (**burda2016importance**), degrades the SNR of the gradient estimates for the inference network, inevitably deteriorating the overall learning process. In short, this behavior manifests because even though increasing K decreases the standard deviation of the gradient estimates, it decreases the magnitude of the true gradient faster, such that the *relative variance increases*.

Our results suggest that it may be best to use distinct objectives for learning the generative and inference networks, or that when using the same target, it should take into account the needs of both networks. Namely, while tighter bounds are typically better for training the generative network, looser bounds often lead to better training of the inference network. Based on these insights, we introduce three new algorithms: the PIWAE, the MIWAE, and the CIWAE. Each of these include IWAE as a special case and are based on the same set of importance weights, but use these weights in different ways to ensure a higher SNR for the inference network.

We demonstrate that our new algorithms can produce inference networks more closely representing the true posterior than IWAE, while matching the training of the generative network, or potentially even improving it in the case of PIWAE. Even when treating the IWAE objective itself as the measure of performance, all our algorithms are able to demonstrate clear improvements over IWAE.

7.2 Background and Notation

Let x be an $\mathcal{X}$-valued random variable defined via a process involving an unobserved $\mathcal{Z}$-valued random variable z with joint density $p_\theta(x, z)$. Direct maximum likelihood estimation of θ is generally intractable if $p_\theta(x, z)$ is a deep generative model due to the marginalization of z. A common strategy is to instead optimize a variational lower bound on $\log p_\theta(x)$, defined via an auxiliary inference model $q_\phi(z|x)$:

$$\begin{aligned} \text{ELBO}_{\text{VAE}}(\theta, \phi, x) &\coloneqq \int q_\phi(z|x) \log \frac{p_\theta(x, z)}{q_\phi(z|x)} \, \mathrm{d}z \\ &= \log p_\theta(x) - \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x)). \end{aligned} \tag{7.1}$$

Typically, q_ϕ is parameterized by a neural network, for which the approach is known as the VAE (**kingma2014auto**; **rezende2014stochastic**). Optimization is performed with stochastic gradient ascent (SGA) using unbiased estimates of $\nabla_{\theta,\phi} \mathrm{ELBO}_{\mathrm{VAE}}(\theta, \phi, x)$. If q_ϕ is reparameterizable, then given a reparameterized sample $z \sim q_\phi(z|x)$, the gradients $\nabla_{\theta,\phi}(\log p_\theta(x, z) - \log q_\phi(z|x))$ can be used for the optimization.

The VAE objective places a harsh penalty on mismatch between $q_\phi(z|x)$ and $p_\theta(z|x)$; optimizing jointly in θ, ϕ can confound improvements in $\log p_\theta(x)$ with reductions in the KL (**turner2011two**). Thus, research has looked to develop bounds that separate the tightness of the bound from the expressiveness of the class of q_ϕ. For example, the IWAE objectives (**burda2016importance**), which we denote as $\mathrm{ELBO}_{\mathrm{IS}}(\theta, \phi, x)$, are a family of bounds defined by

$$\begin{aligned} Q_{\mathrm{IS}}(z_{1:K}|x) &\coloneqq \textstyle\prod_{k=1}^{K} q_\phi(z_k|x), \\ \hat{Z}_{\mathrm{IS}}(z_{1:K}, x) &\coloneqq \frac{1}{K} \textstyle\sum_{k=1}^{K} \frac{p_\theta(x, z_k)}{q_\phi(z_k|x)}, \\ \mathrm{ELBO}_{\mathrm{IS}}(\theta, \phi, x) &\coloneqq \int Q_{\mathrm{IS}}(z_{1:K}|x) \log \hat{Z}_{\mathrm{IS}}(z_{1:K}, x) \, \mathrm{d}z_{1:K} \end{aligned} \tag{7.2}$$

$\leq \log p_\theta(x)$. The IWAE objectives generalize the VAE objective ($K = 1$ corresponds to the VAE) and the bounds become strictly tighter as K increases **burda2016importance**. When the family of q_ϕ contains the true posteriors, the global optimum parameters $\{\theta^*, \phi^*\}$ are independent of K, see e.g. **le2017auto**. Nonetheless, except for the most trivial models, it is not usually the case that q_ϕ contains the true posteriors, and **burda2016importance** provide strong empirical evidence that setting $K > 1$ leads to significant empirical gains over the VAE in terms of learning the generative model.

Optimizing tighter bounds is usually empirically associated with better models p_θ in terms of marginal likelihood on held out data. Other related approaches extend this to sequential Monte Carlo (SMC) (**maddison2017filtering**; **le2017auto**; **naesseth2017variational**) or change the lower bound that is optimized to reduce the bias (**li2016renyi**; **bamler2017perturbative**). A second, unrelated, approach is to tighten the bound by improving the expressiveness of q_ϕ (**salimans_markov_2015**;

tran_variational_2015; **rezende_variational_2015**; **kingma2016improving**; **maaloe_auxiliary_2016**; **ranganath2016hierarchical**). In this work, we focus on the former, algorithmic, approaches to tightening bounds.

7.3 Assessing the Signal-to-Noise Ratio of the Gradient Estimators

Because it is not feasible to analytically optimize these ELBOs in complex models, the effectiveness of any particular choice of ELBO is linked to our ability to numerically solve the resulting optimization problem. This motivates us to examine the effect K has on the variance and magnitude of the gradient estimates of IWAE for the two networks. More generally, we study IWAE gradient estimators constructed as the average of M estimates, each built from K independent particles. We present a result characterizing the asymptotic signal-to-noise ratio in M and K. For the standard case of $M = 1$, our result shows that the signal-to-noise ratio of the reparameterization gradients of the inference network for the IWAE decreases with rate $O(1/\sqrt{K})$.

As estimating the ELBO requires a Monte Carlo estimation of an expectation over z, we have two sample sizes to tune for the estimate: the number of samples M used for Monte Carlo estimation of the ELBO and the number of importance samples K used in the bound construction. Here M does not change the true value of $\nabla_{\theta,\phi}\,\text{ELBO}$, only our variance in estimating it, while changing K changes the ELBO itself, with larger K leading to tighter bounds (**burda2016importance**). Presuming that reparameterization is possible, we can express our gradient estimate in the general form

$$\Delta_{M,K} := \frac{1}{M}\sum_{m=1}^{M} \nabla_{\theta,\phi} \log \frac{1}{K}\sum_{k=1}^{K} w_{m,k}, \tag{7.3}$$
$$\text{where } w_{m,k} = \frac{p_\theta(z_{m,k}, x)}{q_\phi(z_{m,k}|x)} \text{ and } z_{m,k} \sim q_\phi(z_{m,k}|x).$$

Thus, for a fixed budget of $T = MK$ samples, we have a family of estimators with the cases $K = 1$ and $M = 1$ corresponding respectively to the VAE and IWAE

objectives. We will use $\Delta_{M,K}(\theta)$ to refer to gradient estimates with respect to θ and $\Delta_{M,K}(\phi)$ for those with respect to ϕ.

Variance is not always a good barometer for the effectiveness of a gradient estimation scheme; estimators with small expected values need proportionally smaller variances to be estimated accurately. In the case of IWAE, when changes in K simultaneously affect both the variance and expected value, the quality of the estimator for learning can actually *worsen* as the variance decreases. To see why, consider the marginal likelihood estimates $\hat{Z}_{m,K} = \sum_{k=1}^{K} w_{m,k}$. Because these become exact (and thus independent of the proposal) as $K \to \infty$, it must be the case that $\lim_{K\to\infty} \Delta_{M,K}(\phi) = 0$. Thus as K becomes large, the expected value of the gradient must decrease along with its variance, such that the variance relative to the problem scaling need not actually improve.

To investigate this formally, we introduce the signal-to-noise-ratio (SNR), defining it to be the absolute value of the expected estimate scaled by its standard deviation:

$$\text{SNR}_{M,K}(\theta) = \left|\mathbb{E}\left[\Delta_{M,K}(\theta)\right] / \sigma\left[\Delta_{M,K}(\theta)\right]\right| \tag{7.4}$$

where $\sigma[\cdot]$ denotes the standard deviation of a random variable. The SNR is defined separately on each dimension of the parameter vector and similarly for $\text{SNR}_{M,K}(\phi)$. It provides a measure of the relative accuracy of the gradient estimates. Though a high SNR does not always indicate a good SGA scheme (as the target objective itself might be poorly chosen), a low SNR is always problematic as it indicates that the gradient estimates are dominated by noise: if SNR $\to 0$ then the estimates become completely random. We are now ready to state our main theoretical result: $\text{SNR}_{M,K}(\theta) = O(\sqrt{MK})$ and $\text{SNR}_{M,K}(\phi) = O(\sqrt{M/K})$.

Theorem 1. *Assume that when $M = K = 1$, the expected gradients; the variances of the gradients; and the first four moments of $w_{1,1}$, $\nabla_\theta w_{1,1}$, and $\nabla_\phi w_{1,1}$ are all finite and the variances are also non-zero. Then the signal-to-noise ratios of the*

gradient estimates converge at the following rates

$$\text{SNR}_{M,K}(\theta) = \tag{7.5}$$
$$\sqrt{M}\left|\frac{\sqrt{K}\,\nabla_\theta Z - \frac{1}{2Z\sqrt{K}}\nabla_\theta\left(\frac{\text{Var}[w_{1,1}]}{Z^2}\right) + O\left(\frac{1}{K^{3/2}}\right)}{\sqrt{\mathbb{E}\left[w_{1,1}^2\left(\nabla_\theta \log w_{1,1} - \nabla_\theta \log Z\right)^2\right]} + O\left(\frac{1}{K}\right)}\right|$$
$$\text{SNR}_{M,K}(\phi) = \sqrt{M}\left|\frac{\nabla_\phi \text{Var}\left[w_{1,1}\right] + O\left(\frac{1}{K}\right)}{2Z\sqrt{K}\,\sigma\left[\nabla_\phi w_{1,1}\right] + O\left(\frac{1}{\sqrt{K}}\right)}\right| \tag{7.6}$$

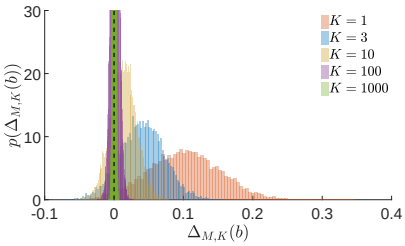
where $Z := p_\theta(x)$ is the true marginal likelihood.

Proof. We give an intuitive demonstration of the result here and provide a formal proof in Appendix 5.A. The effect of M on the SNR follows from using the law of large numbers on the random variable $\nabla_{\theta,\phi} \log \hat{Z}_{m,K}$. Namely, the overall expectation is independent of M and the variance reduces at a rate $O(1/M)$. The effect of K is more complicated but is perhaps most easily seen by noting that **burda2016importance**

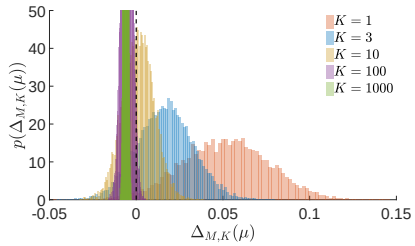
$$\nabla_{\theta,\phi} \log \hat{Z}_{m,K} = \sum_{k=1}^{K} \frac{w_{m,k}}{\sum_{\ell=1}^{K} w_{m,k}} \nabla_{\theta,\phi} \log\left(w_{m,k}\right),$$

such that $\nabla_{\theta,\phi} \log \hat{Z}_{m,K}$ can be interpreted as a self-normalized importance sampling estimate. We can, therefore, invoke the known result (see e.g. **hesterberg1988advances**) that the bias of a self-normalized importance sampler converges at a rate $O(1/K)$ and the standard deviation at a rate $O(1/\sqrt{K})$. We thus see that the SNR converges at a rate $O((1/K)/(1/\sqrt{K})) = O(1/\sqrt{K})$ if the asymptotic gradient is 0 and $O((1)/(1/\sqrt{K})) = O(\sqrt{K})$ otherwise, giving the convergence rates in the ϕ and θ cases respectively. □

The implication of these rates is that increasing M is monotonically beneficial to the SNR for both θ and ϕ, but that increasing K is beneficial to the former and detrimental to the latter. We emphasize that this means the SNR for the IWAE inference network gets worse as we increase K: this is not just an opportunity cost from the fact that we could have increased M instead, increasing the total number of samples used in the estimator actually worsens the SNR!



(a) IWAE inference network gradient estimates



(b) IWAE generative network gradient estimates

Figure 7.3.1: Histograms of gradient estimates $\Delta_{M,K}$ for the generative network and the inference network using the IWAE ($M = 1$) objective with different values of K.

7.3.1 Asymptotic Direction

Even though the magnitude of the inference network gradient estimates diminish with increasing K, the *direction* of the true gradient still converges to a fixed vector. Namely, we have as an intermediary result from deriving (5.6) that

$$\mathbb{E}\left[\Delta_{M,K}(\phi)\right] = -\frac{\nabla_\phi \mathrm{Var}\left[w_{1,1}\right]}{2KZ^2} + O\left(\frac{1}{K^2}\right), \tag{7.7}$$

and we see that expected gradient points in the direction of $-\nabla_\phi \mathrm{Var}\left[w_{1,1}\right]$ as $K \to \infty$. This direction is rather interesting: it implies that as $K \to \infty$, the optimal ϕ is that which minimizes the variance of the weights. This is well known to be the optimal importance sampling distribution in terms of estimating the marginal likelihood (**mcbook**). Given that the role of the inference network during training is to estimate the marginal likelihood, this is thus arguably exactly what we want to optimize for. As such, this result, which complements those of **cremer2017reinterpreting**, suggests that increasing K provides a preferable target in terms of the direction of the true inference network gradients. We thus see that there is a trade-off with the fact that increasing K also diminishes the SNR, reducing the estimates to pure noise if K is set too high. In the absence of other factors, there may thus be a "sweet-spot" for setting K.

7.3.2 Multiple Data Points

Typically when training deep generative models, one does not optimize a single ELBO but instead its average over multiple data points, i.e.

$$\mathcal{J}(\theta, \phi) := \frac{1}{N} \sum\nolimits_{n=1}^{N} \mathrm{ELBO}_{\mathrm{IS}}(\theta, \phi, x^{(n)}). \tag{7.8}$$

Our results extend to this setting because the z are drawn independently for each $x^{(n)}$, so

$$\mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N}\Delta_{M,K}^{(n)}\right]=\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\Delta_{M,K}^{(n)}\right], \tag{7.9}$$

$$\mathrm{Var}\left[\frac{1}{N}\sum_{n=1}^{N}\Delta_{M,K}^{(n)}\right]=\frac{1}{N^2}\sum_{n=1}^{N}\mathrm{Var}\left[\Delta_{M,K}^{(n)}\right]. \tag{7.10}$$

We thus also see that if we are using mini-batches such that N is a chosen parameter and the $x^{(n)}$ are drawn from the empirical data distribution, then the SNRs of $\bar{\Delta}_{N,M,K} := \frac{1}{N}\sum_{n=1}^{N}\Delta_{M,K}^{(n)}$ scales as $\sqrt{N}$, i.e. $\mathrm{SNR}_{N,M,K}(\theta) = O(\sqrt{NMK})$ and $\mathrm{SNR}_{N,M,K}(\phi) = O(\sqrt{NM/K})$. Therefore increasing N has the same ubiquitous benefit as increasing M. In the rest of the paper, we will implicitly be considering the SNRs for $\bar{\Delta}_{N,M,K}$, but will omit the dependency on N to simplify the notation.

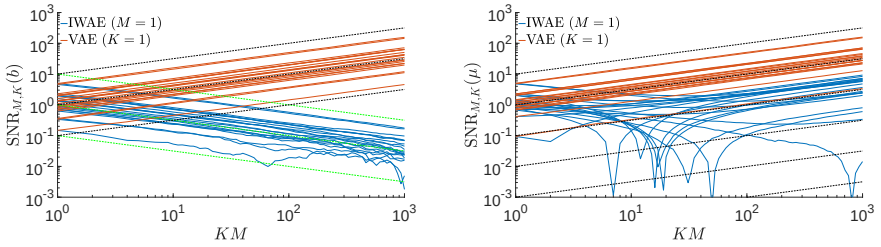
7.4 Empirical Confirmation

Our convergence results hold exactly in relation to M (and N) but are only asymptotic in K due to the higher order terms. Therefore their applicability should be viewed with a healthy degree of skepticism in the small K regime. With this in mind, we now present empirical support for our theoretical results and test how well they hold in the small K regime using a simple Gaussian model, for which we can analytically calculate the ground truth.

Consider a family of generative models with $\mathbb{R}^D$–valued latent variables z and observed variables x:

$$z \sim \mathcal{N}(z;\mu,I), \qquad\qquad x|z \sim \mathcal{N}(x;z,I), \tag{7.11}$$

which is parameterized by $\theta := \mu$. Let the inference network be parameterized by $\phi = (A,b),\ A \in \mathbb{R}^{D\times D},\ b \in \mathbb{R}^D$ where $q_\phi(z|x) = \mathcal{N}(z;Ax+b,\frac{2}{3}I)$. Given a dataset $(x^{(n)})_{n=1}^N$, we can analytically calculate the optimum of our target $\mathcal{J}(\theta,\phi)$ as explained in Appendix 5.B, giving $\theta^* := \mu^* = \frac{1}{N}\sum_{n=1}^{N}x^{(n)}$ and $\phi^* := (A^*,b^*)$, where $A^* = I/2$ and $b^* = \mu^*/2$. Though this will not be the case in general, for this particular problem, the optimal proposal is independent of K. However, the expected gradients for the inference network still change with K.



(a) Convergence of SNR for inference network

(b) Convergence of SNR for generative network

Figure 7.4.1: Convergence of signal-to-noise ratios of gradient estimates with increasing M and K. Different lines correspond to different dimensions of the parameter vectors. Shown in blue is the IWAE where we keep $M = 1$ fixed and increase K. Shown in red is the VAE where $K = 1$ is fixed and we increase M. The black and green dashed lines show the expected convergence rates from our theoretical results, representing gradients of $1/2$ and $-1/2$ respectively.

To conduct our investigation, we randomly generated a synthetic dataset from the model with $D = 20$ dimensions, $N = 1024$ data points, and a true model parameter value μ_{true} that was itself randomly generated from a unit Gaussian, i.e. $\mu_{\text{true}} \sim \mathcal{N}(\mu_{\text{true}}; 0, I)$. We then considered the gradient at a random point in the parameter space close to optimum (we also consider a point far from the optimum in Appendix 5.C.3). Namely each dimension of each parameter was randomly offset from its optimum value using a zero-mean Gaussian with standard deviation 0.01. We then calculated empirical estimates of the ELBO gradients for IWAE, where $M = 1$ is held fixed and we increase K, and for VAE, where $K = 1$ is held fixed and we increase M. In all cases we calculated 10^4 such estimates and used these samples to provide empirical estimates for, amongst other things, the mean and standard deviation of the estimator, and thereby an empirical estimate for the SNR.

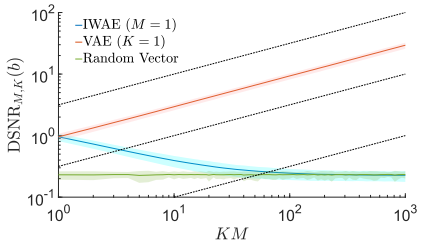
We start by examining the qualitative behavior of the different gradient estimators as K increases as shown in Figure 5.3.1. This shows histograms of the IWAE gradient estimators for a single parameter of the inference network (left) and generative network (right). We first see in Figure 5.3.1a that as K increases, both the magnitude and the standard deviation of the estimator decrease for the inference network, with the former decreasing faster. This matches the qualitative behavior of our theoretical result, with the SNR ratio diminishing as K increases.

In particular, the probability that the gradient is positive or negative becomes roughly equal for larger values of K, meaning the optimizer is equally likely to increase as decrease the inference network parameters at the next iteration. By contrast, for the generative network, IWAE converges towards a non-zero gradient, such that, even though the SNR initially decreases with K, it then rises again, with a very clear gradient signal for $K = 1000$.

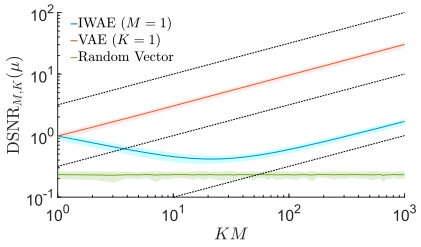
To provide a more rigorous analysis, we next directly examine the convergence of the SNR. Figure 5.4.1 shows the convergence of the estimators with increasing M and K. The observed rates for the inference network (Figure 5.4.1a) correspond to our theoretical results, with the suggested rates observed all the way back to $K = M = 1$. As expected, we see that as M increases, so does $\text{SNR}_{M,K}(b)$, but as K increases, $\text{SNR}_{M,K}(b)$ reduces.

In Figure 5.4.1b, we see that the theoretical convergence for $\text{SNR}_{M,K}(\mu)$ is again observed exactly for variations in M, but a more unusual behavior is seen for variations in K, where the SNR initially decreases before starting to increase again for large enough K, eventually exhibiting behavior consistent with the theoretical result for large enough K. The driving factor for this is that here $\mathbb{E}[\Delta_{M,\infty}(\mu)]$ has a smaller magnitude than (and opposite sign to) $\mathbb{E}[\Delta_{M,1}(\mu)]$ (see Figure 5.3.1b). If we think of the estimators for all values of K as biased estimates for $\mathbb{E}[\Delta_{M,\infty}(\mu)]$, we see from our theoretical results that this bias decreases faster than the standard deviation. Consequently, while the magnitude of this bias remains large compared to $\mathbb{E}[\Delta_{M,\infty}(\mu)]$, it is the predominant component in the true gradient and we see similar SNR behavior as in the inference network.

Note that this does not mean that the estimates are getting worse for the generative network. As we increase K our bound is getting tighter and our estimates closer to the true gradient for the target that we actually want to optimize $\nabla_\mu \log Z$. See Appendix 5.C.2 for more details. As we previously discussed, it is also the case that increasing K could be beneficial for the inference network even if it reduces the SNR by improving the direction of the expected gradient. However, as we will now show, the SNR is, for this problem, the dominant effect for the inference network.

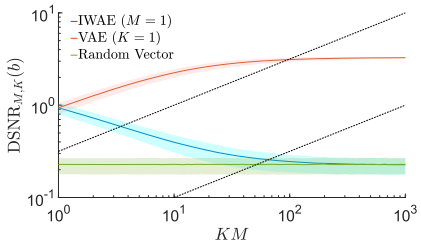


(a) Convergence of DSNR for inference network

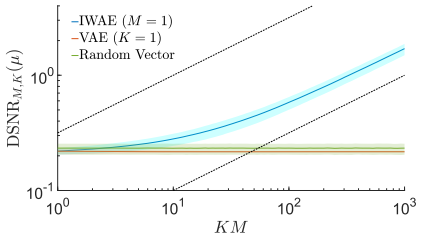


(b) Convergence of DSNR for generative network

Figure 7.4.2: Convergence of the directional SNR of gradients estimates with increasing M and K. The solid lines show the estimated DSNR and the shaded regions the interquartile range of the individual ratios. Also shown for reference is the DSNR for a randomly generated vector where each component is drawn from a unit Gaussian.



(a) Convergence of DSNR for inference network



(b) Convergence of DSNR for generative network

Figure 7.4.3: Convergence of the DSNR when the target gradient is taken as $u = \mathbb{E}[\Delta_{1,1000}]$. Conventions as per Figure 5.4.2.

7.4.1 Directional Signal-to-Noise Ratio

As a reassurance that our chosen definition of the SNR is appropriate for the problem at hand and to examine the effect of multiple dimensions explicitly, we now also consider an alternative definition of the SNR that is similar (though distinct) to that used in **roberts2009signal**. We refer to this as the "directional" SNR (DSNR). At a high-level, we define the DSNR by splitting each gradient estimate into two component vectors, one parallel to the true gradient and one perpendicular, then taking the expectation of ratio of their magnitudes. More precisely, we define $u = \mathbb{E}[\Delta_{M,K}] / \|\mathbb{E}[\Delta_{M,K}]\|_2$ as being the true normalized gradient direction and then the DSNR as

$$\text{DSNR}_{M,K} = \mathbb{E}\left[\frac{\|\Delta_{\|}\|_2}{\|\Delta_{\mathbf{o}_t}\|_2}\right] \quad \text{where} \tag{7.12}$$
$$\Delta_{\|} = \left(\Delta_{M,K}^T u\right) u \quad \text{and} \quad \Delta_{\mathbf{o}_t} = \Delta_{M,K} - \Delta_{\|}.$$

The DSNR thus provides a measure of the expected proportion of the gradient that will point in the true direction. For perfect estimates of the gradients, then DSNR $\to \infty$, but unlike the SNR, arbitrarily bad estimates do not have DSNR $= 0$ because even random vectors will have a component of their gradient in the true direction.

The convergence of the DSNR is shown in Figure 5.4.2, for which the true normalized gradient u has been estimated empirically, noting that this varies with K. We see a similar qualitative behavior to the SNR, with the gradients of IWAE for the inference network degrading to having the same directional accuracy as drawing a random vector. Interestingly, the DSNR seems to be following the same asymptotic convergence behavior as the SNR for both networks in M (as shown by the dashed lines), even though we have no theoretical result to suggest this should occur.

As our theoretical results suggest that the direction of the true gradients correspond to targeting an improved objective as K increases, we now examine whether this or the changes in the SNR is the dominant effect. To this end, we repeat our calculations for the DSNR but take u as the target direction of the gradient for $K = 1000$. This provides a measure of how varying M and K affects the quality of the gradient directions as biased estimators for $\mathbb{E}\left[\Delta_{1,1000}\right] / \|\mathbb{E}\left[\Delta_{1,1000}\right]\|_2$. As shown in Figure 5.4.3, increasing K is still detrimental for the inference network by this metric, even though it brings the expected gradient estimate closer to the target gradient. By contrast, increasing K is now monotonically beneficial for the generative network. Increasing M leads to initial improvements for the inference network before plateauing due to the bias of the estimator. For the generative network, increasing M has little impact, with the bias being the dominant factor throughout. Though this metric is not an absolute measure of performance of the SGA scheme, e.g. because high bias may be more detrimental than high variance, it is nonetheless a powerful result in suggesting that increasing K can be detrimental to learning the inference network.

7.5 New Estimators

Based on our theoretical results, we now introduce three new algorithms that address the issue of diminishing SNR for the inference network. Our first, MIWAE, is exactly equivalent to the general formulation given in (5.3), the distinction from previous approaches coming from the fact that it takes both $M > 1$ and $K > 1$. The motivation for this is that because our inference network SNR increases as $O(\sqrt{M/K})$, we should be able to mitigate the issues increasing K has on the SNR by also increasing M. For fairness, we will keep our overall budget $T = MK$ fixed, but we will show that given this budget, the optimal value for M is often not 1. In practice, we expect that it will often be beneficial to increase the mini-batch size N rather than M for MIWAE; as we showed in Section 5.3.2 this has the same effect on the SNR. Nonetheless, MIWAE forms an interesting reference method for testing our theoretical results and, as we will show, it can offer improvements over IWAE for a given N.

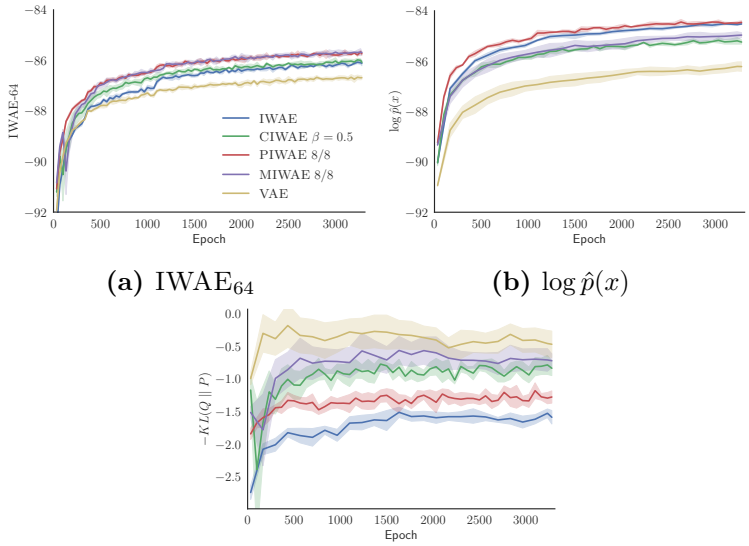
Our second algorithm, CIWAE uses a convex combination of the IWAE and VAE bounds, namely

$$\text{ELBO}_{\text{CIWAE}} = \beta\, \text{ELBO}_{\text{VAE}} + (1-\beta)\, \text{ELBO}_{\text{IWAE}} \tag{7.13}$$

where $\beta \in [0, 1]$ is a combination parameter. It is trivial to see that $\text{ELBO}_{\text{CIWAE}}$ is a lower bound on the log marginal that is tighter than the VAE bound but looser than the IWAE bound. We then employ the following estimator

$$\Delta^{\text{C}}_{K,\beta} = \nabla_{\theta,\phi} \left(\beta \frac{1}{K} \sum_{k=1}^{K} \log w_k + (1-\beta) \log \left(\frac{1}{K} \sum_{k=1}^{K} w_k \right) \right) \tag{7.14}$$

where we use the same w_k for both terms. The motivation for CIWAE is that, if we set β to a relatively small value, the objective will behave mostly like IWAE, except when the expected IWAE gradient becomes very small. When this happens, the VAE component should "take-over" and alleviate SNR issues: the asymptotic SNR of $\Delta^{\text{C}}_{K,\beta}$ for ϕ is $O(\sqrt{MK})$ because the VAE component has non-zero expectation in the limit $K \to \infty$.



(a) IWAE_{64} **(b)** $\log \hat{p}(x)$

(c) $-\text{KL}(Q_\phi(z|x)||P_\theta(z|x))$

Figure 7.5.1: Convergence of evaluation metrics on the test set with increased training time. All lines show mean ± standard deviation over 4 runs with different random initializations. Larger values are preferable for each plot.

Metric	IWAE	PIWAE $(4,16)$	PIWAE $(8,8)$	MIWAE $(4,16)$	MIWAE $(8,8)$	CIWAE $\beta = 0.05$	CIWAE $\beta = 0.5$	VA
IWAE-64	-86.11 ± 0.10	$\mathbf{-85.68 \pm 0.06}$	$\mathbf{-85.74 \pm 0.07}$	$\mathbf{-85.60 \pm 0.07}$	$\mathbf{-85.69 \pm 0.04}$	-85.91 ± 0.11	-86.08 ± 0.08	$-86.69 \pm$
$\log \hat{p}(x)$	$\mathbf{-84.52 \pm 0.02}$	$\mathbf{-84.40 \pm 0.17}$	$\mathbf{-84.46 \pm 0.06}$	$\mathbf{-84.56 \pm 0.05}$	-84.97 ± 0.10	$\mathbf{-84.57 \pm 0.09}$	-85.24 ± 0.08	$-86.21 \pm$
$-\text{KL}(Q\|\|P)$	-1.59 ± 0.10	-1.27 ± 0.18	-1.28 ± 0.09	$\mathbf{-1.04 \pm 0.08}$	$\mathbf{-0.72 \pm 0.11}$	-1.34 ± 0.14	$\mathbf{-0.84 \pm 0.11}$	$\mathbf{-0.47 \pm}$

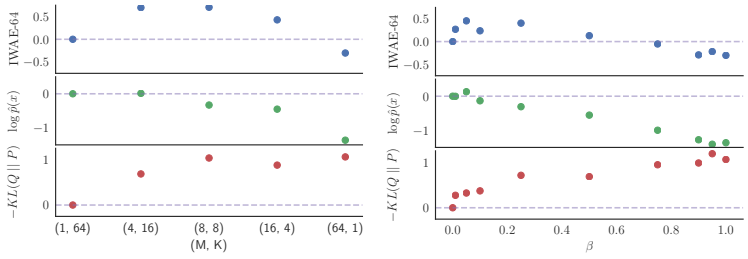
Table 7.5.1: Mean final test set performance ± standard deviation over 4 runs. Numbers in brackets indicate (M, K). The best result is shown in red, while bold results are not statistically significant to best result at the 5% level of a Welch's t-test.

Our results suggest that what is good for the generative network, in terms of setting K, is often detrimental for the inference network. It is therefore natural to question whether it is sensible to always use the same target for both the inference and generative networks. Motivated by this, our third method, PIWAE, uses the IWAE target when training the generative network, but the MIWAE target for training the inference network. We thus have

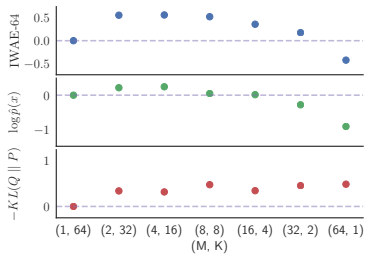
$$\Delta^{\text{P}}_{M,K}(\theta) = \nabla_\theta \log \frac{1}{K} \sum_{k=1}^{K} w_k \tag{7.15a}$$

$$\Delta^{\text{P}}_{M,K}(\phi) = \frac{1}{M} \sum_{m=1}^{M} \nabla_\phi \log \frac{1}{L} \sum_{\ell=1}^{L} w_{m,\ell} \tag{7.15b}$$

where we will generally set $K = ML$ so that the same weights can be used for both gradients.



(a) Comparing MIWAE and IWAE **(b)** Comparing CIWAE and IWAE



(c) Comparing PIWAE and IWAE

Figure 7.5.2: Test set performance of MIWAE, CIWAE, and PIWAE relative to IWAE in terms of the IWAE-64 (top), $\log \hat{p}(x)$ (middle), and $-\mathrm{KL}(Q_\phi(z|x)||P_\theta(z|x))$ (bottom) metrics. All dots are the difference in the metric to that of IWAE. Dotted line is the IWAE baseline. Note that in all cases, the far left of the plot correspond to settings equivalent to the IWAE.

7.5.1 Experiments

We now use our new estimators to train deep generative models for the MNIST digits dataset (**lecun1998gradient**). For this, we duplicated the architecture and training schedule outlined in **burda2016importance**. In particular, all networks were trained and evaluated using their stochastic binarization. For all methods we set a budget of $T = 64$ weights in the target estimate for each datapoint in the minibatch.

To assess different aspects of the training performance, we consider three different metrics: $\mathrm{ELBO}_{\mathrm{IWAE}}$ with $K = 64$, $\mathrm{ELBO}_{\mathrm{IWAE}}$ with $K = 5000$, and the latter of these minus the former. All reported metrics are evaluated on the test data.

The motivation for the $\mathrm{ELBO}_{\mathrm{IWAE}}$ with $K = 64$ metric, denoted as IWAE-64, is that this is the target used for training the IWAE and so if another method does better on this metric than the IWAE, this is a clear indicator that SNR issues of the IWAE estimator have degraded its performance. In fact, this would demonstrate that, from a practical perspective, using the IWAE estimator is sub-optimal, even if our explicit aim is to optimize the IWAE bound. The second metric, $\mathrm{ELBO}_{\mathrm{IWAE}}$ with

$K = 5000$, denoted $\log \hat{p}(x)$, is used as a surrogate for estimating the log marginal likelihood and thus provides an indicator for fidelity of the learned generative model. The third metric is an estimator for the divergence implicitly targeted by the IWAE. Namely, as shown by **le2017auto**, the $\text{ELBO}_{\text{IWAE}}$ can be interpreted as

$$\text{ELBO}_{\text{IWAE}} = \log p_\theta(x) - \text{KL}(Q_\phi(z|x)||P_\theta(z|x)) \tag{7.16}$$

$$\text{where} \quad Q_\phi(z|x) := \prod_{k=1}^{K} q_\phi(z_k|x), \quad \text{and} \tag{7.17}$$

$$P_\theta(z|x) := \frac{1}{K}\sum_{k=1}^{K} \frac{\prod_{\ell=1}^{K} q_\phi(z_\ell|x)}{q_\phi(z_k|x)} p_\theta(z_k|x). \tag{7.18}$$

Thus we can estimate $\text{KL}(Q_\phi(z|x)||P_\theta(z|x))$ using $\log \hat{p}(x) -$ IWAE-64, to provide a metric for divergence between the inference network and the proposal network. We use this instead of $\text{KL}(q_\phi(z|x)||p_\theta(z|x))$ because the latter can be deceptive metric for the inference network fidelity. For example, it tends to prefer $q_\phi(z|x)$ that cover only one of the posterior modes, rather than encompassing all of them. As we showed in Section 5.3.1, the implied target of the true gradients for the inference network improves as K increases and so $\text{KL}(Q_\phi(z|x)||P_\theta(z|x))$ should be a more reliable metric of inference network performance.

Figure 5.5.1 shows the convergence of these metrics for each algorithm. Here we have considered the middle value for each of the parameters, namely $K = M = 8$ for PIWAE and MIWAE, and $\beta = 0.5$ for CIWAE. We see that PIWAE and MIWAE both comfortably outperformed, and CIWAE slightly outperformed, IWAE in terms of IWAE-64 metric, despite IWAE being directly trained on this target. In terms of $\log \hat{p}(x)$, PIWAE gave the best performance, followed by IWAE. For the KL, we see that the VAE performed best followed by MIWAE, with IWAE performing the worst. We note here that the KL is not an exact measure of the inference network performance as it also depends on the generative model. As such, the apparent superior performance of the VAE may be because it produces a simpler model, as per the observations of **burda2016importance**, which in turn is easier to learn an inference network for. Critically though, PIWAE improves this metric whilst also improving generative network performance, such that this reasoning no

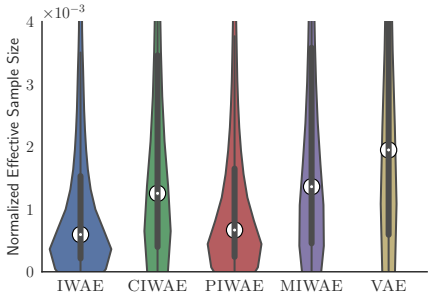


Figure 7.5.3: Violin plots of ESS estimates for each image of MNIST, normalized by the number of samples drawn. A violin plot uses a kernel density plot on each side – thicker means more MNIST images whose q_ϕ achieves that ESS.

longer applies. Similar behavior is observed for MIWAE and CIWAE for different parameter settings (see Appendix 5.D).

We next considered tuning the parameters for each of our algorithms as shown in Figure 5.5.2, for which we look at the final metric values after training. Table 5.5.1 further summarizes the performance for certain selected parameter settings. For MIWAE we see that as we increase M, the $\log \hat{p}(x)$ metric gets worse, while the KL gets better. The IWAE-64 metric initially increases with M, before reducing again from $M = 16$ to $M = 64$, suggesting that intermediate values for M (i.e. $M \neq 1$, $K \neq 1$) give a better trade-off. For PIWAE, similar behavior to MIWAE is seen for the IWAE-64 and KL metrics. However, unlike for MIWAE, we see that $\log \hat{p}(x)$ initially increases with M, such that PIWAE provides uniform improvement over IWAE for the $M = 2, 4, 8$, and 16 cases. CIWAE exhibits similar behavior in increasing β as increasing M for MIWAE, but there appears to be a larger degree of noise in the evaluations, while the optimal value of β, though non-zero, seems to be closer to IWAE than for the other algorithms.

As an additional measure of the performance of the inference network that is distinct to any of the training targets, we also considered the effective sample size (ESS) **mcbook** for the fully trained networks, defined as

$$\text{ESS} = (\textstyle\sum_{k=1}^{K} w_k)^2 / \sum_{k=1}^{K} w_k^2. \tag{7.19}$$

The ESS is a measure of how many unweighted samples would be equivalent to the weighted sample set. A low ESS indicates that the inference network is struggling to perform effective inference for the generative network. The results, given in

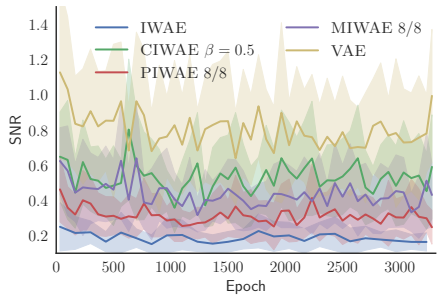


Figure 7.5.4: SNR of inference network weights during training. All lines are mean ± standard deviation over 20 randomly chosen weights per layer.

Figure 5.5.3, show that the ESSs for CIWAE, MIWAE, and the VAE were all significantly larger than for IWAE and PIWAE, with IWAE giving a particularly poor ESS.

Our final experiment looks at the SNR values for the inference networks during training. Here we took a number of different neural network gradient weights at different layers of the network and calculated empirical estimates for their SNRs at various points during the training. We then averaged these estimates over the different network weights, the results of which are given in Figure 5.5.4. This clearly shows the low SNR exhibited by the IWAE inference network, suggesting that our results from the simple Gaussian experiments carry over to the more complex neural network domain.

7.6 Conclusions

We have provided theoretical and empirical evidence that algorithmic approaches to increasing the tightness of the ELBO independently to the expressiveness of the inference network can be detrimental to learning by reducing the signal-to-noise ratio of the inference network gradients. Experiments on a simple latent variable model confirmed our theoretical findings. We then exploited these insights to introduce three estimators, PIWAE, MIWAE, and CIWAE and showed that each can deliver improvements over IWAE, even when the metric used for this assessment is the IWAE target itself. In particular, each was able to deliver improvement in the training of the inference network, while maintaining the quality of the learned generative network.

Whereas MIWAE and CIWAE mostly allow for balancing the requirements of the inference and generative networks, PIWAE appears to be able to offer simultaneous

improvements to both, with the improved training of the inference network having a knock-on effect on the generative network. Key to achieving this is, is its use of separate targets for the two networks, opening up interesting avenues for future work.

Acknowledgments

TR and YWT are supported in part by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no. 617071. TAL is supported by a Google studentship, project code DF6700. MI is supported by the UK EPSRC CDT in Autonomous Intelligent Machines and Systems. CJM is funded by a DeepMind Scholarship. FW is supported under DARPA PPAML through the U.S. AFRL under Cooperative Agreement FA8750-14-2-0006, Sub Award number 61160290-111668.

Appendix

7.A Proof of SNR Convergence Rates

Theorem 1. *Assume that when $M = K = 1$, the expected gradients; the variances of the gradients; and the first four moments of $w_{1,1}$, $\nabla_\theta w_{1,1}$, and $\nabla_\phi w_{1,1}$ are all finite and the variances are also non-zero. Then the signal-to-noise ratios of the gradient estimates converge at the following rates*

$$\text{SNR}_{M,K}(\theta) = \sqrt{M}\left|\frac{\sqrt{K}\,\nabla_\theta Z - \frac{1}{2Z\sqrt{K}}\nabla_\theta\left(\frac{\text{Var}[w_{1,1}]}{Z^2}\right) + O\left(\frac{1}{K^{3/2}}\right)}{\sqrt{\mathbb{E}\left[w_{1,1}^2\left(\nabla_\theta \log w_{1,1} - \nabla_\theta \log Z\right)^2\right]} + O\left(\frac{1}{K}\right)}\right| \tag{7.20}$$

$$\text{SNR}_{M,K}(\phi) = \sqrt{M}\left|\frac{\nabla_\phi \text{Var}\left[w_{1,1}\right] + O\left(\frac{1}{K}\right)}{2Z\sqrt{K}\,\sigma\left[\nabla_\phi w_{1,1}\right] + O\left(\frac{1}{\sqrt{K}}\right)}\right| \tag{7.21}$$

where $Z := p_\theta(x)$ is the true marginal likelihood.

Proof. We start by considering the variance of the estimators. We will first exploit the fact that each $\hat{Z}_{m,K}$ is independent and identically distributed and then apply Taylor's theorem[1] to $\log \hat{Z}_{m,K}$ about Z, using $R_1(\cdot)$ to indicate the remainder term, as follows.

$$\begin{aligned}
&M \cdot \text{Var}\left[\Delta_{M,K}\right] = \text{Var}\left[\Delta_{1,K}\right] = \text{Var}\left[\nabla_{\theta,\phi}\left(\log Z + \frac{\hat{Z}_{1,K} - Z}{Z} + R_1\left(\hat{Z}_{1,K}\right)\right)\right] \\
=&\text{Var}\left[\nabla_{\theta,\phi}\left(\frac{\hat{Z}_{1,K} - Z}{Z} + R_1\left(\hat{Z}_{1,K}\right)\right)\right] \\
=&\mathbb{E}\left[\left(\nabla_{\theta,\phi}\left(\frac{\hat{Z}_{1,K} - Z}{Z} + R_1\left(\hat{Z}_{1,K}\right)\right)\right)^2\right] - \left(\mathbb{E}\left[\nabla_{\theta,\phi}\left(\frac{\hat{Z}_{1,K} - Z}{Z} + R_1\left(\hat{Z}_{1,K}\right)\right)\right]\right)^2
\end{aligned}$$

[1] This approach follows similar lines to the derivation of nested Monte Carlo convergence bounds in **rainforth2017thesis**; **rainforth2017opportunities**; **fort2017mcmc** and the derivation of the mean squared error for self-normalized importance sampling, see e.g. **hesterberg1988advances**.

$$
\begin{aligned}
=&\mathbb{E}\left[\left(\frac{1}{K}\sum_{k=1}^{K}\frac{Z\nabla_{\theta,\phi}w_{1,k}-w_{1,k}\nabla_{\theta,\phi}Z}{Z^2}+\nabla_{\theta,\phi}R_1\left(\hat{Z}_{1,K}\right)\right)^2\right]-\left(\nabla_{\theta,\phi}\mathbb{E}\left[\frac{\hat{Z}_{1,K}-Z}{Z}\right]^{0}+\mathbb{E}\left[\nabla_{\theta,\phi}R_1\left(\hat{Z}_{1,K}\right)\right]\right)^2\\
=&\frac{1}{KZ^4}\mathbb{E}\left[\left(Z\nabla_{\theta,\phi}w_{1,1}-w_{1,1}\nabla_{\theta,\phi}Z\right)^2\right]+\mathrm{Var}\left[\nabla_{\theta,\phi}R_1\left(\hat{Z}_{1,K}\right)\right]\\
&+2\mathbb{E}\left[\left(\nabla_{\theta,\phi}R_1\left(\hat{Z}_{1,K}\right)\right)\left(\frac{1}{K}\sum_{k=1}^{K}\frac{Z\nabla_{\theta,\phi}w_{1,k}-w_{1,k}\nabla_{\theta,\phi}Z}{Z^2}\right)\right]
\end{aligned}
$$

Now we have by the mean-value form of the remainder that for some $\tilde{Z}$ between Z and $\hat{Z}_{1,K}$

$$R_1\left(\hat{Z}_{1,K}\right)=-\frac{\left(\hat{Z}_{1,K}-Z\right)^2}{2\tilde{Z}^2}$$

and therefore

$$\nabla_{\theta,\phi}R_1\left(\hat{Z}_{1,K}\right)=-\frac{\tilde{Z}\left(\hat{Z}_{1,K}-Z\right)\nabla_{\theta,\phi}\left(\hat{Z}_{1,K}-Z\right)-\left(\hat{Z}_{1,K}-Z\right)^2\nabla_{\theta,\phi}\tilde{Z}}{\tilde{Z}^3}.$$

It follows that the $\nabla_{\theta,\phi}R_1\left(\hat{Z}_{1,K}\right)$ terms are dominated as each of $\left(\hat{Z}_{1,K}-Z\right)\nabla_{\theta,\phi}\left(\hat{Z}_{1,K}-Z\right)$ and $\left(\hat{Z}_{1,K}-Z\right)^2$ vary with the square of the estimator error, whereas other comparable terms vary only with the unsquared difference. The assumptions on moments of the weights and their derivatives further guarantee that these terms are finite. More precisely, we have $\tilde{Z}=Z+\alpha(\hat{Z}_{1,K}-Z)$ for some $0<\alpha<1$ where $\nabla_{\theta,\phi}\alpha$ must be bounded with probability 1 as $K\to\infty$ to maintain our assumptions. It follows that $\nabla_{\theta,\phi}R_1(\hat{Z}_{1,K})=O((\hat{Z}_{1,K}-Z)^2)$ and thus that

$$\mathrm{Var}\left[\Delta_{M,K}\right]=\frac{1}{MKZ^4}\mathbb{E}\left[\left(Z\nabla_{\theta,\phi}w_{1,1}-w_{1,1}\nabla_{\theta,\phi}Z\right)^2\right]+\frac{1}{M}O\left(\frac{1}{K^2}\right)\tag{7.22}$$

using the fact that the third and fourth order moments of a Monte Carlo estimator both decrease at a rate $O(1/K^2)$.

Considering now the expected gradient estimate and again using Taylor's theorem, this time to a higher number of terms,

$$
\begin{aligned}
\mathbb{E}\left[\Delta_{M,K}\right]=\mathbb{E}\left[\Delta_{1,K}\right]&=\mathbb{E}\left[\Delta_{1,K}-\nabla_{\theta,\phi}\log Z\right]+\nabla_{\theta,\phi}\log Z\\
&=\nabla_{\theta,\phi}\mathbb{E}\left[\log Z+\frac{\hat{Z}_{1,K}-Z}{Z}-\frac{\left(\hat{Z}_{1,K}-Z\right)^2}{2Z^2}+R_2\left(\hat{Z}_{1,K}\right)-\log Z\right]+\nabla_{\theta,\phi}\log Z
\end{aligned}
$$

$$= -\frac{1}{2}\nabla_{\theta,\phi}\mathbb{E}\left[\left(\frac{\hat{Z}_{1,K}-Z}{Z}\right)^2\right] + \nabla_{\theta,\phi}\mathbb{E}\left[R_2\left(\hat{Z}_{1,K}\right)\right] + \nabla_{\theta,\phi}\log Z$$

$$= -\frac{1}{2}\nabla_{\theta,\phi}\left(\frac{\mathrm{Var}\left[\hat{Z}_{1,K}\right]}{Z^2}\right) + \nabla_{\theta,\phi}\mathbb{E}\left[R_2\left(\hat{Z}_{1,K}\right)\right] + \nabla_{\theta,\phi}\log Z$$

$$= -\frac{1}{2K}\nabla_{\theta,\phi}\left(\frac{\mathrm{Var}\left[w_{1,1}\right]}{Z^2}\right) + \nabla_{\theta,\phi}\mathbb{E}\left[R_2\left(\hat{Z}_{1,K}\right)\right] + \nabla_{\theta,\phi}\log Z. \tag{7.23}$$

Using a similar process as in variance case, it is now straightforward to show that $\nabla_{\theta,\phi}\mathbb{E}\left[R_2\left(\hat{Z}_{1,K}\right)\right] = O(1/K^2)$, which is thus similarly dominated (also giving us (5.7)).

Finally, by combing (5.22) and (5.23) and noting that $\sqrt{\frac{A}{K}+\frac{B}{K^2}} = \frac{A}{\sqrt{K}} + \frac{B}{2AK^{3/2}} + O\left(\frac{1}{K^{(5/2)}}\right)$ we have

$$\mathrm{SNR}_{M,K}(\theta) = \left|\frac{\nabla_\theta \log Z - \frac{1}{2K}\nabla_\theta\left(\frac{\mathrm{Var}[w_{1,1}]}{Z^2}\right) + O\left(\frac{1}{K^2}\right)}{\sqrt{\frac{1}{MKZ^4}\mathbb{E}\left[\left(Z\nabla_\theta w_{1,1} - w_{1,1}\nabla_\theta Z\right)^2\right] + \frac{1}{M}O\left(\frac{1}{K^2}\right)}}\right| \tag{7.24}$$

$$= \sqrt{M}\left|\frac{Z^2\sqrt{K}\left(\nabla_\theta \log Z - \frac{1}{2K}\nabla_\theta\left(\frac{\mathrm{Var}[w_{1,1}]}{Z^2}\right)\right) + O\left(\frac{1}{K^{3/2}}\right)}{\sqrt{\mathbb{E}\left[\left(Z\nabla_\theta w_{1,1} - w_{1,1}\nabla_\theta Z\right)^2\right] + O\left(\frac{1}{K}\right)}}\right| \tag{7.25}$$

$$= \sqrt{M}\left|\frac{\sqrt{K}\,\nabla_\theta Z - \frac{1}{2Z\sqrt{K}}\nabla_\theta\left(\frac{\mathrm{Var}[w_{1,1}]}{Z^2}\right) + O\left(\frac{1}{K^{3/2}}\right)}{\sqrt{\mathbb{E}\left[w_{1,1}^2\left(\nabla_\theta \log w_{1,1} - \nabla_\theta \log Z\right)^2\right] + O\left(\frac{1}{K}\right)}}\right| = O\left(\sqrt{MK}\right). \tag{7.26}$$

For ϕ, then because $\nabla_\phi Z = 0$, we instead have

$$\mathrm{SNR}_{M,K}(\phi) = \sqrt{M}\left|\frac{\nabla_\phi \mathrm{Var}\left[w_{1,1}\right] + O\left(\frac{1}{K}\right)}{2Z\sqrt{K}\;\sigma\left[\nabla_\phi w_{1,1}\right] + O\left(\frac{1}{\sqrt{K}}\right)}\right| = O\left(\sqrt{\frac{M}{K}}\right) \tag{7.27}$$

and we are done. □

7.B Derivation of Optimal Parameters for Gaussian Experiment

To derive the optimal parameters for the Gaussian experiment we first note that

$$\mathcal{J}(\theta,\phi) = \frac{1}{N}\log\prod_{n=1}^{N} p_\theta(x^{(n)}) - \frac{1}{N}\sum_{n=1}^{N} D_{\mathrm{KL}}Q_\phi(z_{1:K}|x^{(n)})P_\theta(z_{1:K}|x^{(n)}) \quad \text{where}$$

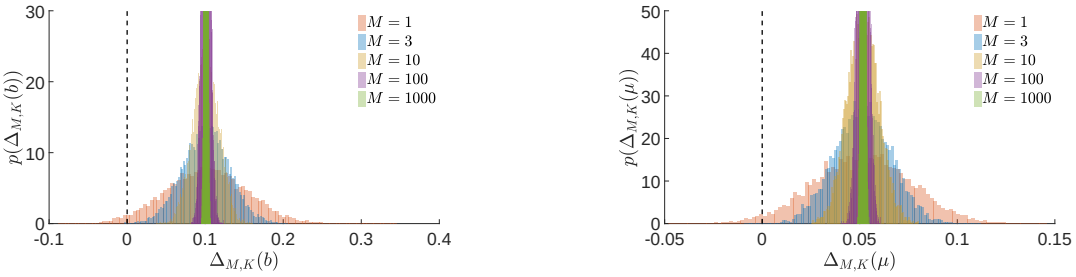
$$P_\theta(z_{1:K}|x^{(n)}) = \frac{1}{K}\sum_{k=1}^{K} q_\phi(z_1|x^{(n)})\ldots q_\phi(z_{k-1}|x^{(n)})p_\theta(z_k|x^{(n)})q_\phi(z_{k+1}|x^{(n)})\ldots q_\phi(z_K|x^{(n)}),$$

$Q_\phi(z_{1:K}|x^{(n)})$ is as per (5.2) and the form of the KL is taken from **le2017auto**. Next, we note that ϕ only controls the mean of the proposal so, while it is not possible to drive the KL to zero, it will be minimized for any particular θ when the means of $q_\phi(z|x^{(n)})$ and $p_\theta(z|x^{(n)})$ are the same. Furthermore, the corresponding minimum possible value of the KL is independent of θ and so we can calculate the optimum pair (θ^*, ϕ^*) by first optimizing for θ and then choosing the matching ϕ. The optimal θ maximizes $\log \prod_{n=1}^N p_\theta(x^{(n)})$, giving $\theta^* := \mu^* = \frac{1}{N}\sum_{n=1}^N x^{(n)}$. As we straightforwardly have $p_\theta(z|x^{(n)}) = \mathcal{N}(z; \left(x^{(n)} + \mu\right)/2, I/2)$, the KL is then minimized when $A = I/2$ and $b = \mu/2$, giving $\phi^* := (A^*, b^*)$, where $A^* = I/2$ and $b^* = \mu^*/2$.

7.C Additional Empirical Analysis of SNR

7.C.1 Histograms for VAE

To complete the picture for the effect of M and K on the distribution of the gradients, we generated histograms for the $K = 1$ (i.e. VAE) gradients as M is varied. As shown in Figure 5.C.1a, we see the expected effect from the law of large numbers that the variance of the estimates decreases with M, but not the expected value.



(a) VAE inference network gradient estimates

(b) VAE generative network gradient estimates

Figure 7.C.1: Histograms of gradient estimates $\Delta_{M,K}$ for the generative network and the inference network using the VAE ($K = 1$) objectives with different values of M.

7.C.2 Convergence of RMSE for Generative Network

As explained in the main paper, the SNR is not an entirely appropriate metric for the generative network – a low SNR is still highly problematic, but

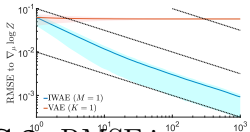


Figure 7.C.2: RMSE in μ gradient estimate to $\nabla_\mu \log Z$

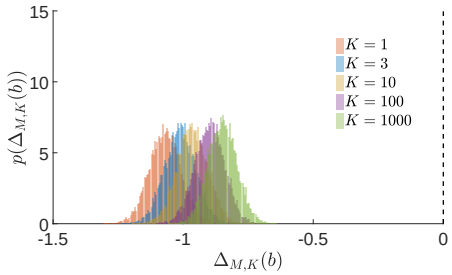
a high SNR does not indicate good

performance. It is thus perhaps better to measure the quality of the gradient estimates for the generative network by looking at the root mean squared error (RMSE) to $\nabla_\mu \log Z$, i.e. $\sqrt{\mathbb{E}\left[\|\Delta_{M,K} - \nabla_\mu \log Z\|_2^2\right]}$. The convergence of this RMSE is shown in Figure 5.C.2 where the solid lines are the RMSE estimates using 10^4 runs and the shaded regions show the interquartile range of the individual estimates. We see that increasing M in the VAE reduces the variance of the estimates but has negligible effect on the RMSE due to the fixed bias. On the other hand, we see that increasing K leads to a monotonic improvement, initially improving at a rate $O(1/K)$ (because the bias is the dominating term in this region), before settling to the standard Monte Carlo convergence rate of $O(1/\sqrt{K})$ (shown by the dashed lines).

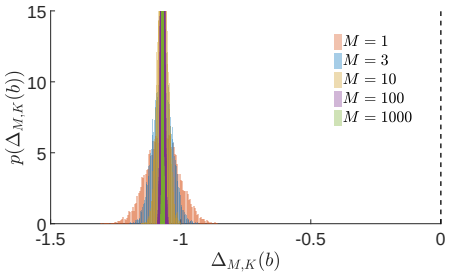
7.C.3 Experimental Results for High Variance Regime

We now present empirical results for a case where our weights are higher variance. Instead of choosing a point close to the optimum by offsetting parameters with a standard deviation of 0.01, we instead offset using a standard deviation of 0.5. We further increased the proposal covariance to I to make it more diffuse. This is now a scenario where the model is far from its optimum and the proposal is a very poor match for the model, giving very high variance weights.

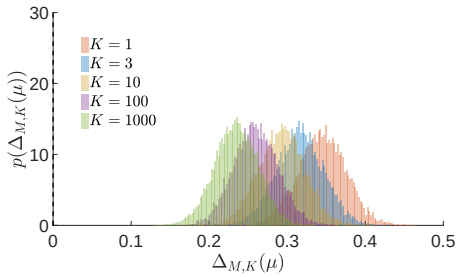
We see that the behavior is the same for variation in M, but somewhat distinct for variation in K. In particular, the SNR and DSNR only decrease slowly with K for the inference network, while increasing K no longer has much benefit for the SNR of the inference network. It is clear that, for this setup, the problem is very far from the asymptotic regime in K such that our theoretical results no longer directly apply. Nonetheless, the high-level effect observed is still that the SNR of the inference network deteriorates, albeit slowly, as K increases.



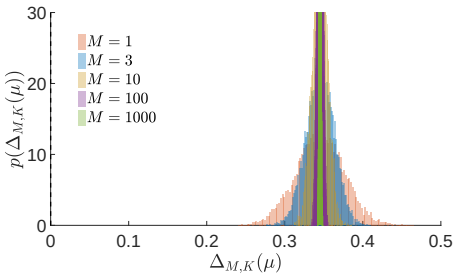
(a) IWAE inference network gradient estimates



(b) VAE inference network gradient estimates

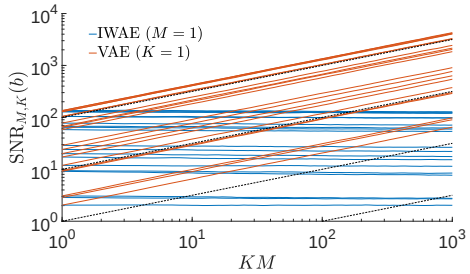


(c) IWAE generative network gradient estimates

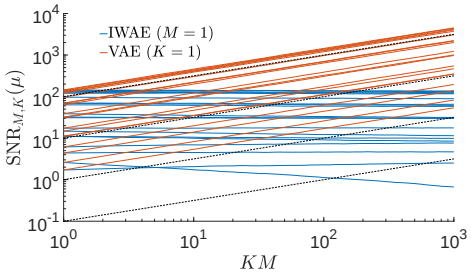


(d) VAE generative network gradient estimates

Figure 7.C.3: Histograms of gradient estimates as per Figure 5.3.1.



(a) Convergence of SNR for inference network

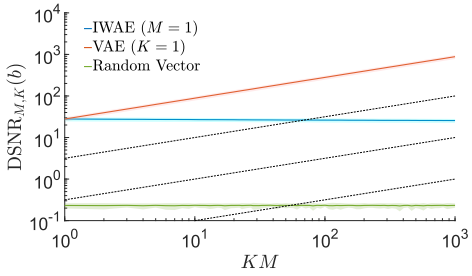


(b) Convergence of SNR for generative network

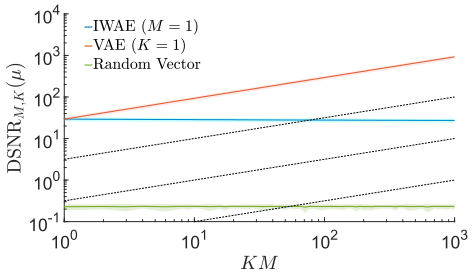
Figure 7.C.4: Convergence of signal-to-noise ratios of gradient estimates as per Figure 5.4.1.

7.D Convergence of Deep Generative Model for Alternative Parameter Settings

Figure 5.D.1 shows the convergence of the introduced algorithms under different settings to those shown in Figure 5.5.1. Namely we consider $M = 4, K = 16$ for PIWAE and MIWAE and $\beta = 0.05$ for CIWAE. These settings all represent tighter bounds than those of the main paper. Similar behavior is seen in terms of the IWAE-64 metric for all algorithms. PIWAE produced similar mean behavior for all metrics,

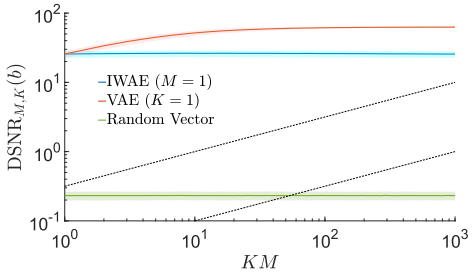


(a) Convergence of DSNR for inference network

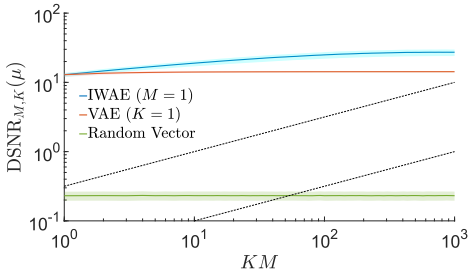


(b) Convergence of DSNR for generative network

Figure 7.C.5: Convergence of directional signal-to-noise ratio of gradients estimates as per Figure 5.4.2.



(a) Convergence of DSNR for inference network



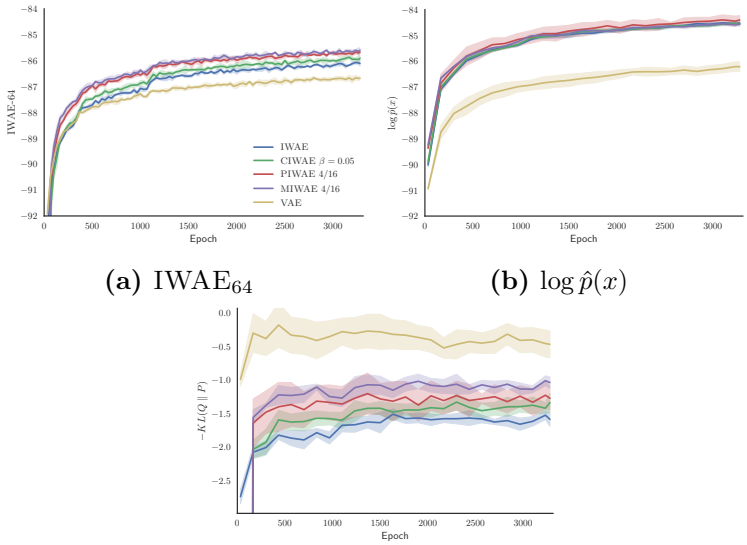
(b) Convergence of DSNR for generative network

Figure 7.C.6: Convergence of directional signal-to-noise ratio of gradient estimates where the true gradient is taken as $\mathbb{E}\left[\Delta_{1,1000}\right]$ as per Figure 5.4.3.

though the variance was noticeably increased for $\log \hat{p}(x)$. For CIWAE and MIWAE, we see that the parameter settings represent an explicit trade-off between the generative network and the inference network: $\log \hat{p}(x)$ was noticeably increased for both, matching that of IWAE, while $-\mathrm{KL}(Q_\phi(z|x)||P_\theta(z|x))$ was reduced. Critically, we see here that, as observed for PIWAE in the main paper, MIWAE and CIWAE are able to match the generative model performance of IWAE whilst improving the KL metric, indicating that they have learned better inference networks.

7.E Convergence of Toy Gaussian Problem

We finish by assessing the effect of the outlined changes in the quality of the gradient estimates on the final optimization for our toy Gaussian problem. Figure 5.E.1 shows the convergence of running Adam (**kingma2014adam**) to optimize μ, A, and b. This suggests that the effects observed predominantly transfer to the overall



(a) IWAE_{64} **(b)** $\log \hat{p}(x)$

(c) $-\text{KL}(Q_\phi(z|x)||P_\theta(z|x))$

Figure 7.D.1: Convergence of different evaluation metrics for each method. Plotting conventions as per Figure 5.5.1.

optimization problem. Interestingly, setting $K = 1$ and $M = 1000$ gave the best performance on learning not only the inference network parameters, but also the generative network parameters.

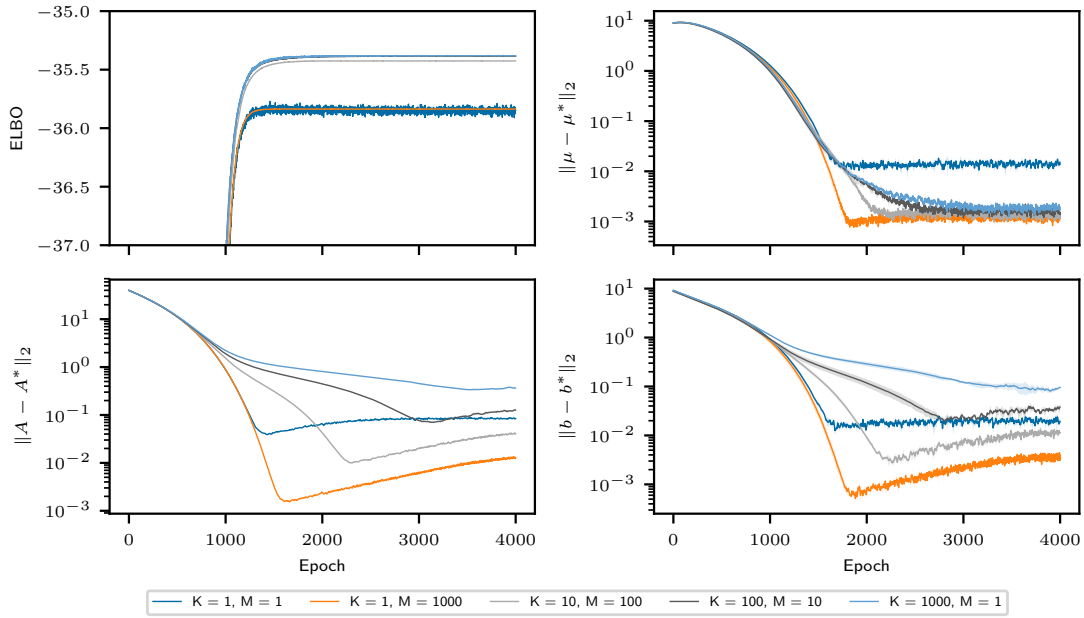


Figure 7.E.1: Convergence of optimization for different values of K and M. *(Top, left)* ELBO_{IS} during training (note this represents a different metric for different K). *(Top, right)* L_2 distance of the generative network parameters from the true maximizer. *(Bottom)* L_2 distance of the inference network parameters from the true maximizer. Plots show means over 3 repeats with ± 1 standard deviation. Optimization is performed using the Adam algorithm with all parameters initialized by sampling from the uniform distribution on $[1.5, 2.5]$.

8

Revisiting Reweighted Wake-Sleep for Models with Stochastic Control Flow

Abstract

SCFMs are a class of generative models that involve branching on choices from discrete random variables. Amortized gradient-based learning of SCFMs is challenging as most approaches targeting discrete variables rely on their continuous relaxations—which can be intractable in SCFMs, as branching on relaxations requires evaluating *all* (exponentially many) branching paths. Tractable alternatives mainly combine REINFORCE with complex control-variate schemes to improve the variance of naïve estimators. Here, we revisit the reweighted wake-sleep (RWS) (**bornschein2015reweighted**) algorithm, and through extensive evaluations, show that it outperforms current state-of-the-art methods in learning SCFMs. Further, in contrast to the importance-weighted auto-encoder, we observe that RWS learns better models *and* inference networks with increasing numbers of particles. Our results suggest that RWS is a competitive, often preferable, alternative for learning SCFMs.

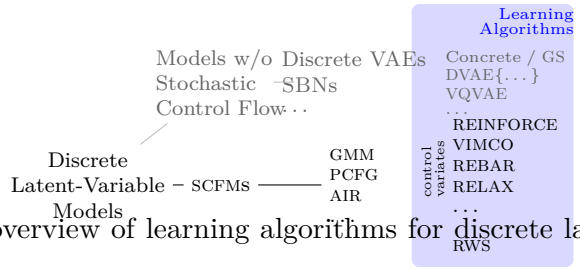


Figure 8.1.1: An overview of learning algorithms for discrete latent-variable models, with focus on SCFMs.

8.1 Introduction

SCFMs describe generative models that employ branching (i.e., the use of `if` / `else` / `cond` statements) on choices from discrete random variables. Recent years have seen such models gain relevance, particularly in the domain of deep probabilistic programming (**siddharth2017learning**; **bingham2019pyro**; **tran2017deep**; **vandemeent2018intro**), which allows combining neural networks with generative models expressing arbitrarily complex control flow. SCFMs are encountered in a wide variety of tasks including tracking and prediction (**neiswanger2014dependent**; **kosiorek2018sequential**), clustering (**rasmussen2000infinite**), topic modeling (**blei2003latent**), model structure learning (**adams2010learning**), counting (**eslami2016attend**), attention (**xu2015show**), differentiable data structures (**graves2014neural**; **graves2016hybrid**; **grefenstette2015learning**), speech & language modeling (**juang1991hidden**; **chater2006probabilistic**), and concept learning (**kemp2006learning**; **lake2018emergence**).

While a variety of approaches for amortized gradient-based learning (targeting model ELBO) exist for models using discrete random variables, the majority rely on continuous relaxations of the discrete variables (**rolfe2016dvae**; **vahdat2018dvaepp**; **vahdat2018dvaehash**; **oord2017neural**; **maddison2017concrete**; **jang2017categorical**), enabling gradient computation through reparameterization (**kingma2014auto**; **rezende2014stochastic**). The models used by these approaches typically do not involve any control flow on discrete random choices, instead choosing to feed choices from their continuous relaxations directly into a neural network, thereby facilitating the required learning.

In contrast, SCFMs do not lend themselves to continuous-relaxation-based approaches due to the explicit branching requirement on choices from the discrete

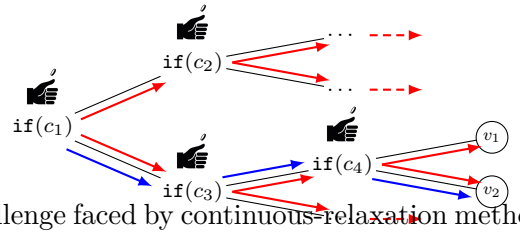


Figure 8.1.2: The challenge faced by continuous-relaxation methods on SCFMs—requiring exploration of all branches, in contrast to exploring only one branch at a time. Stochastic control flow proceeds through discrete choices (c_i) yielding values (v_i).

variables. Consider for example a simple SCFM—a two-mixture GMM. Computing the ELBO for this model involves choosing mixture identity (Bernoulli). Since a sample from a relaxed variable denotes a point on the surface of a probability simplex (e.g. [0.2, 0.8] for a Bernoulli random variable), instead of its vertices (0 or 1), computing the ELBO would need evaluation of both branches, weighting the resulting computation under each branch appropriately. This process can very quickly become intractable for more complex SCFMs, as it requires evaluation of *all* possible branches in the computation, of which there may be *exponentially* many, as illustrated in Fig. 6.1.2.

Alternatives to continuous-relaxation methods mainly involve the use of the IWAE (**burda2016importance**) framework, employing the REINFORCE (**williams1992simple**) gradient estimator, combined with control-variate schemes (**mnih2014neural**; **mnih2016variational**; **gu2016muprop**; **tucker2017rebar**; **grathwohl2018backpropagatio** to help decrease the variance of the naïve estimator. Although this approach ameliorates the problem with continuous relaxations in that it does not require evaluation of all branches, it has other drawbacks. Firstly, with more particles, the IWAE estimator adversely impacts inference-network quality, consequently impeding model learning (**rainforth2018tighter**). Secondly, its practical efficacy can still be limited due to high variance and the requirement to design and optimize a separate neural network (c.f. Section 6.4.3).

Having characterized the class of models we are interested in (c.f. Fig. 6.1.1), and identified a range of current approaches (along with their characteristics) that might apply to such models, we revisit RWS (**bornschein2015reweighted**). Comparing extensively with state-of-the-art methods for learning in SCFMs, we demonstrate its

efficacy in learning better generative models and inference networks, using lower variance gradient estimators, over a range of computational budgets. To this end, we first review state-of-the-art methods for learning deep generative models with discrete latent variables (Section 6.2). We then revisit RWS (Section 6.3) and present an extensive evaluation of these methods (Section 6.4) on i) a PCFG model on sentences, ii) the AIR model (**eslami2016attend**) to perceive and localize multiple MNIST digits, and iii) a pedagogical GMM example that exposes a shortcoming of RWS which we then design a fix for. Our experiments confirm RWS as a competitive, often preferable, alternative for learning SCFMs.

8.2 Background

Consider data $(x^{(n)})_{n=1}^N$ sampled from a true (unknown) generative model $p(x)$, a family of generative models $p_\theta(z, x)$ of latent variable z and observation x parameterized by θ and a family of inference networks $q_\phi(z|x)$ parameterized by ϕ. We aim to learn the generative model by maximizing the marginal likelihood over data: $\theta^* = \arg\max_\theta \frac{1}{N} \sum_{n=1}^N \log p_\theta(x^{(n)})$. Simultaneously, we would like to learn an inference network $q_\phi(z|x)$ that amortizes inference given observation x; i.e., $q_\phi(z|x)$ maps an observation x to an approximation of $p_{\theta^*}(z|x)$. Amortization ensures this function evaluation is cheaper than performing approximate inference of $p_{\theta^*}(z|x)$ from scratch. Our focus here is on such joint learning of generative model and inference network, here referred to as "learning a deep generative model", although we note that other approaches exist that learn the generative model (**goodfellow2014generative**; **mohamed2016learning**) or inference network (**paige2016inference**; **le2017inference**) in isolation.

We begin by reviewing IWAEs (**burda2016importance**) as a general approach for learning deep generative models using stochastic gradient descent (SGD) methods, focusing on generative-model families with discrete latent variables, for which the naïve gradient estimator's high variance impedes learning. We also review control-variate and continuous-relaxation methods for gradient-variance reduction. IWAEs

coupled with such gradient-variance reduction methods are currently the dominant approach for learning deep generative models with discrete latent variables.

8.2.1 Importance Weighted Autoencoder

burda2016importance introduce the IWAE, maximizing the mean ELBOs over data, $\frac{1}{N}\sum_{n=1}^{N}\text{ELBO}_{\text{IS}}^{K}(\theta,\phi,x^{(n)})$, where, for K particles,

$$\text{ELBO}_{\text{IS}}^{K}(\theta,\phi,x)=\mathbb{E}_{Q_\phi(z_{1:K}|x)}\left[\log\left(\frac{1}{K}\sum_{k=1}^{K}w_k\right)\right], \tag{8.1}$$
$$Q_\phi(z_{1:K}|x)=\prod_{k=1}^{K}q_\phi(z_k|x),\ w_k=\frac{p_\theta(z_k,x)}{q_\phi(z_k|x)}.$$

When $K=1$, this reduces to the VAE (**kingma2014auto**; **rezende2014stochastic**). **burda2016importance** show that $\text{ELBO}_{\text{IS}}^{K}(\theta,\phi,x)$ is a lower bound on $\log p_\theta(x)$ and that increasing K leads to a tighter lower bound. Further, tighter lower bounds arising from increasing K improve learning of the generative model, but impair learning of the inference network (**rainforth2018tighter**), as the signal-to-noise ratio of θ's gradient estimator is $O(\sqrt{K})$ whereas ϕ's is $O(1/\sqrt{K})$. Note that although **tucker2019doubly** solve this for reparameterizable distributions, the issue persists for discrete distributions. Consequently, poor learning of the inference network, beyond a certain point (large K), can actually impair learning of the generative model as well; a finding we explore in Section 6.4.3.

Optimizing the IWAE objective using SGD methods requires unbiased gradient estimators of $\text{ELBO}_{\text{IS}}^{K}(\theta,\phi,x)$ with respect to θ and ϕ (**robbins1951stochastic**). $\nabla_\theta\,\text{ELBO}_{\text{IS}}^{K}(\theta,\phi,x)$ is estimated by evaluating $\nabla_\theta\log\hat{Z}_K$ using samples $z_{1:K}\sim Q_\phi(\cdot|x)$, where $\hat{Z}_K=\frac{1}{K}\sum_{k=1}^{K}w_k$. $\nabla_\phi\text{ELBO}_{\text{IS}}^{K}(\theta,\phi,x)$ is estimated similarly for models with reparameterizable latents, discrete (and other non-reparameterizable) latents require the REINFORCE gradient estimator (**williams1992simple**)

$$g_{\text{REINFORCE}}=\underbrace{\log\hat{Z}_K\nabla_\phi\log Q_\phi(z_{1:K}|x)}_{①}+\underbrace{\nabla_\phi\log\hat{Z}_K}_{②}. \tag{8.2}$$

8.2.2 Continuous Relaxations and Control Variates

Since the gradient estimator in Eq. (6.2) typically suffers from high variance, mainly due to the effect of ①, a number of approaches have been developed to ameliorate the issue. These can be broadly categorized into approaches that directly transform the discrete latent variables (continuous relaxations), or approaches that target improvement of the naïve REINFORCE estimator (control variates).

Continuous Relaxations: Here, discrete variables are transformed to enable reparameterization (**kingma2014auto**; **rezende2014stochastic**), helping reduce gradient-estimator variance. Approaches span the Gumbel distribution (**maddison2017concrete**; **jang2017categorical**), spike-and-X transforms (**rolfe2016dvae**), overlapping exponentials (**vahdat2018dvaepp**), and generalized overlapping exponentials for tighter bounds (**vahdat2018dvaehash**).

Besides difficulties inherent to such methods, such as tuning temperature parameters, or the suitability of undirected Boltzmann machine priors, these methods are not well suited for learning SCFMs as they generate samples on the surface of a probability simplex rather than its vertices. For example, sampling from a transformed Bernoulli distribution yields samples of the form $[\alpha, (1-\alpha)]$ rather than simply 0 or 1—the latter form required for branching. With relaxed samples, as illustrated in Fig. 6.1.2, one would need to execute *all* the exponentially many discrete-variable driven branches in the model, weighting each branch appropriately—something that can quickly become infeasible for even moderately complex models. However, for purposes of comparison, for relatively simple SCFMs, one could apply methods involving continuous relaxations, as demonstrated in Section 6.4.3.

Control Variates: Here, approaches build on the REINFORCE estimator for the IWAE ELBO objective, designing control-variate schemes to reduce the variance of the naïve estimator. VIMCO (**mnih2016variational**) eschews designing an explicit control variate, instead exploiting the particle set obtained in IWAE. It

replaces ① with

$$\begin{aligned} \overset{①}{g}_{\text{VIMCO}} &= \sum_{k=1}^{K} (\log \hat{Z}_K - \Upsilon_{-k}) \nabla_\phi \log q_\phi(z_k|x), \\ \Upsilon_{-k} &= \log \frac{1}{K} \left(\exp\left(\frac{1}{K-1} \sum_{\ell \neq k} \log w_\ell \right) + \sum_{\ell \neq k} w_\ell \right) \end{aligned} \tag{8.3}$$

where $\Upsilon_{-k} \perp\!\!\!\perp z_k$ and highly correlated with $\log \hat{Z}_K$.

Finally, assuming z_k is a discrete random variable with C categories[1], RE-BAR (**tucker2017rebar**) and RELAX (**grathwohl2018backpropagation**) improve on **mnih2014neural** and **gu2016muprop**, replacing ① as

$$\begin{aligned} \overset{①}{g}_{\text{RELAX}} = &\left(\log \hat{Z}_K - c_\rho(\tilde{g}_{1:K}) \right) \nabla_\phi \log Q_\phi(z_{1:K}|x) \\ &+ \nabla_\phi c_\rho(g_{1:K}) - \nabla_\phi c_\rho(\tilde{g}_{1:K}), \end{aligned} \tag{8.4}$$

where g_k is a C-dimensional vector of reparameterized Gumbel random variates, z_k is a one-hot argmax function of g_k, and $\tilde{g}_k$ is a vector of reparameterized conditional Gumbel random variates conditioned on z_k. The conditional Gumbel random variates are a form of Rao-Blackwellization used to reduce variance. The control variate c_ρ, parameterized by ρ, is optimized to minimize the gradient variance estimates along with the main ELBO optimization, leading to state-of-the-art performance on, for example, sigmoid belief networks (**neal1992connectionist**). The main difficulty in using this method is choosing a suitable family of c_ρ, as some choices lead to higher variance despite concurrent gradient-variance minimization.

8.3 Revisiting Reweighted Wake-Sleep

Reweighted wake-sleep (RWS) (**bornschein2015reweighted**) comes from a family of algorithms (**hinton1995wake**; **dayan1995helmholtz**) for learning deep generative models, eschewing a single objective over parameters θ and ϕ in favour of individual objectives for each. We review the RWS algorithm and discuss its pros and cons.

[1]The assumption is needed only for notational convenience. However, using more structured latents leads to difficulties in picking the control-variate architecture.

8.3.1 Reweighted Wake-Sleep

Reweighted wake-sleep (RWS) (**bornschein2015reweighted**) is an extension of the wake-sleep algorithm (**hinton1995wake**; **dayan1995helmholtz**) both of which, like IWAE, jointly learn a generative model and an inference network given data. While IWAE targets a single objective, RWS alternates between objectives, updating the generative model parameters θ using a *wake-phase* θ *update* and the inference network parameters ϕ using either a *sleep-* or a *wake-phase* ϕ *update* (or both).

Wake-phase θ update. Given ϕ, θ is updated using an unbiased estimate of $\nabla_\theta - \left(\frac{1}{N}\sum_{n=1}^{N} \text{ELBO}_{\text{IS}}^{K}(\theta, \phi, x^{(n)})\right)$, obtained without reparameterization or control variates, as the sampling distribution $Q_\phi(\cdot|x)$ is independent of θ.[2]

Sleep-phase ϕ update. Here, ϕ is updated to minimize the KL divergence between the posteriors under the generative model and the inference network, averaged over the data distribution of the current generative model

$$\begin{aligned}&\mathbb{E}_{p_\theta(x)}[D_{\text{KL}}(p_\theta(z|x), q_\phi(z|x))] \\ &\qquad = \mathbb{E}_{p_\theta(z,x)}[\log p_\theta(z|x) - \log q_\phi(z|x)].\end{aligned} \tag{8.5}$$

Its gradient, $\mathbb{E}_{p_\theta(z,x)}[-\nabla_\phi \log q_\phi(z|x)]$, is estimated by evaluating $-\nabla_\phi \log q_\phi(z|x)$, where $z, x \sim p_\theta(z, x)$. The estimator's variance can be reduced at a standard Monte Carlo rate by increasing the number of samples of z, x.

Wake-phase ϕ update. Here, ϕ is updated to minimize the KL divergence between the posteriors under the generative model and the inference network, averaged over the true data distribution

$$\begin{aligned}&\mathbb{E}_{p(x)}[D_{\text{KL}}(p_\theta(z|x), q_\phi(z|x))] \\ &\qquad = \mathbb{E}_{p(x)}[\mathbb{E}_{p_\theta(z|x)}[\log p_\theta(z|x) - \log q_\phi(z|x)]].\end{aligned} \tag{8.6}$$

The outer expectation $\mathbb{E}_{p(x)}[\mathbb{E}_{p_\theta(z|x)}[-\nabla_\phi \log q_\phi(z|x)]]$ of the gradient is estimated using a single sample x from the true data distribution $p(x)$, given which, the inner

[2] We assume that the deterministic mappings induced by the parameters θ, ϕ are themselves differentiable, such that they are amenable to gradient-based learning.

expectation is estimated using self-normalized importance sampling with K particles, using $q_\phi(z|x)$ as the proposal distribution. This results in the following estimator

$$\sum_{k=1}^{K} \frac{w_k}{\sum_{\ell=1}^{K} w_\ell} \left(-\nabla_\phi \log q_\phi(z_k|x)\right), \tag{8.7}$$

where, similar to Eq. (6.1), $x \sim p(x)$, $z_k \sim q_\phi(z_k|x)$, and $w_k = p_\theta(z_k, x)/q_\phi(z_k|x)$. Note that Eq. (6.7) is the negative of the second term of the REINFORCE estimator of the IWAE ELBO in Eq. (6.2). The crucial difference between the *wake-phase ϕ update* and the *sleep-phase ϕ update* is that the expectation in Eq. (6.6) is over the *true data distribution* $p(x)$ and the expectation in Eq. (6.5) is under the *current model distribution* $p_\theta(x)$. The former is desirable from the perspective of amortizing inference over data from $p(x)$, and although its estimator is biased, this bias decreases as K increases.

8.3.2 Pros of Reweighted Wake-Sleep

While the gradient update of θ targets the same objective as IWAE, the gradient update of ϕ targets the objective in Eq. (6.5) in the sleep case and Eq. (6.6) in the wake case. This makes RWS a preferable option to IWAE for learning inference networks because the ϕ updates in RWS directly target minimization of the expected KL divergences from the true to approximate posterior. With an increased computational budget, using more Monte Carlo samples in the *sleep-phase ϕ update* case and more particles K in the *wake-phase ϕ update*, we obtain a better estimator of these expected KL divergences. This is in contrast to IWAE, where optimizing $\text{ELBO}_{\text{IS}}^{K}$ targets a KL divergence on an extended sampling space (**le2018autoencoding**) which for $K > 1$ doesn't correspond to a KL divergence between true and approximate posteriors (in any order). Consequently, increasing K in IWAE leads to impaired learning of inference networks (**rainforth2018tighter**).

Moreover, targeting $D_{\text{KL}}(p, q)$ as in RWS can be preferable to targeting $D_{\text{KL}}(q, p)$ as in VAEs. The former encourages *mean-seeking behavior*, having the inference network to put non-zero mass in regions where the posterior has non-zero mass, whereas the latter encourages *mode-seeking behavior*, having the inference network to put

mass on one of the modes of the posterior (**minka2005divergence**). Using the inference network as an IS proposal requires mean-seeking behavior (**owen2013monte**). Moreover, **chatterjee2018sample** show that the number of particles required for IS to accurately approximate expectations of the form $\mathbb{E}_{p(z|x)}[f(z)]$ is directly related to $\exp(D_{\mathrm{KL}}(p, q))$.

8.3.3 Cons of Reweighted Wake-Sleep

While a common criticism of the wake-sleep family of algorithms is the lack of a unifying objective, we have not found any empirical evidence where this is a problem. Perhaps a more relevant criticism is that both the sleep and wake-phase ϕ gradient estimators are biased with respect to $\nabla_\phi \mathbb{E}_{p(x)}[D_{\mathrm{KL}}(p_\theta(z|x), q_\phi(z|x))]$. The bias in the sleep-phase ϕ gradient estimator arises from targeting the expectation under the model rather than the true data distribution, and the bias in the wake-phase ϕ gradient estimator results from estimating the KL divergence using self-normalized IS.

In theory, these biases should not affect the fixed point of optimization (θ^*, ϕ^*) where $p_{\theta^*}(x) = p(x)$ and $q_{\phi^*}(z|x) = p_{\theta^*}(z|x)$. First, if $\theta \to \theta^*$ through the wake-phase θ update, the data distribution bias reduces to zero. Second, although the wake-phase ϕ gradient estimator is biased, it is consistent—with large enough K, convergence of stochastic optimization is theoretically guaranteed on convex objectives and empirically on non-convex objectives (**chen2018stochastic**). Further, this gradient estimator follows the central limit theorem, so its asymptotic variance decreases linearly with K (**owen2013monte**). Thus, using larger K improves learning of the inference network.

In practice, the families of generative models, inference networks, and the data distributions determine which of the biases are more significant. In most of our findings, the bias of the data distribution appears to be the most detrimental. This is due to the fact that initially $p_\theta(x)$ is quite different from $p(x)$, and hence using sleep-phase ϕ updates performs worse than using wake-phase ϕ updates. An exception to this is the PCFG experiment (c.f. Section 6.4.1) where the data distribution bias is not as large and inference using self-normalized IS is extremely difficult.

8.4 Experiments

The IWAE and RWS algorithms have primarily been applied to problems with continuous latent variables and/or discrete latent variables that do not actually induce branching (such as sigmoid belief networks; **neal1992connectionist**). The purpose of the following experiments is to compare RWS to IWAE combined with control variates and continuous relaxations (c.f Section 6.3) on models with conditional branching, and show that it outperform such methods. We empirically demonstrate that increasing the number of particles K can be detrimental in IWAE but advantageous in RWS, as evidenced by achieved ELBOs and average distance between true and amortized posteriors.

In the first experiment, we present learning and amortized inference in a PCFG (**booth1973applying**), an example SCFM where continuous relaxations are inapplicable. We demonstrate that RWS outperforms IWAE with a control variate both in terms of learning and inference. The second experiment focuses on Attend, Infer, Repeat (AIR), the deep generative model of **eslami2016attend**. It demonstrates that RWS leads to better learning of the generative model in a setting with both discrete and continuous latent variables, for modeling a complex visual data domain (c.f. Section 6.4.2). The final experiment involves a GMM (Section 6.4.3), thereby serving as a pedagogical example. It explains the causes of why RWS might be preferable to other methods in more detail. [3]

Notationally, the different variants of RWS will be referred to as WS and WW. The *wake-phase θ update* is always used. We refer to using it in conjunction with the *sleep-phase ϕ update* as WS and using it in conjunction with the *wake-phase ϕ update* as WW. Using both *wake-* and *sleep-phase ϕ updates* doubles the required stochastic sampling while yielding only minor improvements on the models we considered. The number of particles K used for the *wake-phase θ* and *ϕ updates* is always specified, and computation between them is matched so a *wake-phase ϕ update* with batch size B implies a *sleep phase ϕ update* with KB samples.

[3] In Section 6.D, we include additional experiments on sigmoid belief networks which, however, are not SCFMs.

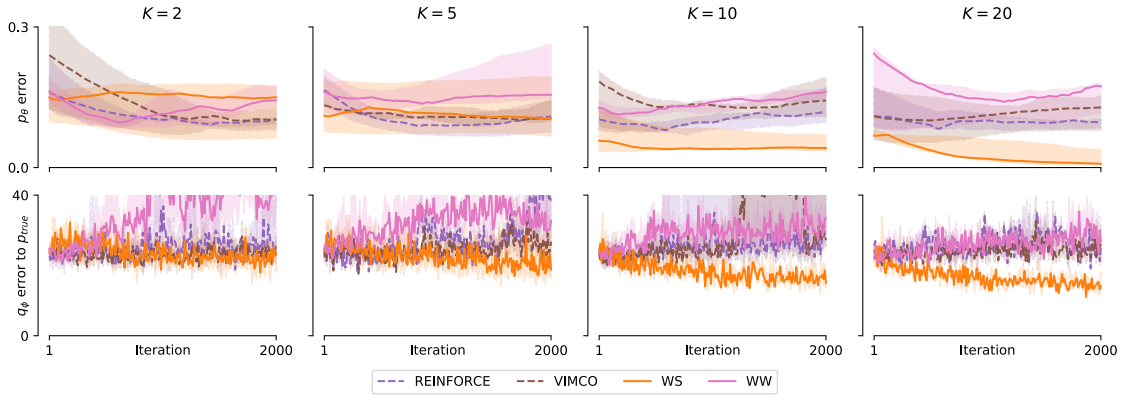


Figure 8.4.1: PCFG training. *(Top)* Quality of the generative model: While all methods have the same gradient update for θ, the performance of WS improves and is the best as K is increased. Other methods, including WW, do not yield significantly better model learning as K is increased, since WS's inference network learns the fastest. *(Bottom)* Quality of the inference network: VIMCO and REINFORCE do not improve with increasing K. WS performs best as K is increased, and while WW's performance improves, the improvement is not as significant. This can be attributed to the data-distribution bias being less significant than the bias coming from self-normalized IS (c.f. Section 6.3.3). Median and interquartile ranges from up to 10 repeats shown (see text).

8.4.1 Probabilistic Context-Free Grammars

In this experiment we learn model parameters and amortize inference in a PCFG (**booth1973applying**). Each discrete latent variable in a PCFG chooses a particular child of a node in a tree. Depending on each discrete choice, the generative model can lead to different future latent variables. A PCFG is an example of an SCFM where continuous relaxations cannot be applied—weighing combinatorially many futures by a continuous relaxation is infeasible and doing so for futures which have infinite latent variables is impossible.

While supervised approaches have recently led to state-of-the-art performance in parsing (**chen2014fast**), PCFGs remain one of the key models for unsupervised parsing (**manning1999foundations**). Learning in a PCFG is typically done via expectation-maximization (**dempster1977maximum**) which uses the inside-outside algorithm (**lari1990estimation**). Inference methods are based on dynamic programming (**younger1967recognition**; **earley1970efficient**) or search (**klein2003parsing**). Applying RWS and IWAE algorithms to PCFGs allows learning from large unlabeled datasets through SGD while inference amortization

ensures linear-time parsing in the number of words in a sentence, at test-time. Moreover, using the inference network as a proposal distribution in IS provides asymptotically exact posteriors if parses are ambiguous.

A PCFG is defined by sets of terminals (or words) $\{t_i\}$, non-terminals $\{n_i\}$, production rules $\{n_i \to \zeta_j\}$ with ζ_j a sequence of terminals and non-terminals, probabilities for each production rule such that $\sum_j P(n_i \to \zeta_j) = 1$ for each n_i, and a start symbol n_1. Consider the *Astronomers* PCFG given in **manning1999foundations** (c.f. Section 6.A). A parse tree z is obtained by recursively applying the production rules until there are no more non-terminals. For example, a parse tree (S (NP astronomers) (VP (V saw) (NP stars))) is obtained by applying the production rules as follows:

$$\begin{aligned} \mathrm{S} &\xrightarrow{1.0} \mathrm{NP\,VP} \xrightarrow{0.1} \text{astronomers}\,\mathrm{VP} \xrightarrow{0.7} \text{astronomers}\,\mathrm{V\,NP} \\ &\xrightarrow{1.0} \text{astronomers saw}\,\mathrm{NP} \xrightarrow{0.18} \text{astronomers saw stars}, \end{aligned}$$

where the probability $p(z)$ is obtained by multiplying the corresponding production probabilities as indicated on top of the arrows. The likelihood of a PCFG, $p(x|z)$, is 1 if the sentence x matches the sentence produced by z (in this case "astronomers saw stars") and 0 otherwise. One can easily construct infinitely long z by choosing productions which contain non-terminals, for example: $\mathrm{S} \to \mathrm{NP\,VP} \to \mathrm{NP\,PP\,VP} \to \mathrm{NP\,PP\,PP\,VP} \to \cdots$.

We learn the production probabilities of the PCFG and an inference network computing the conditional distribution of a parse tree given a sentence. The architecture of the inference network is the same as described in (**le2017inference**) except the input to the RNN consists only of the sentence embedding, previous sample embedding, and an address embedding. Each word is represented as a one-hot vector and the sentence embedding is obtained through another RNN. Instead of a hard $\{0, 1\}$ likelihood which can make learning difficult, we use a relaxation, $p(x|z) = \exp(-L(x, s(z))^2)$, where L is the Levenshtein distance and $s(z)$ is the sentence produced by z. Using the Levenshtein distance can also be interpreted as an instance of approximate Bayesian computation (**sisson2018handbook**).

Training sentences are obtained by sampling from the *astronomers* PCFG with the true production probabilities.

We run WW, WS, VIMCO and REINFORCE ten times for $K \in \{2, 5, 10, 20\}$, with batch size $B = 2$, using the Adam optimizer (**kingma2015adam**) with default hyperparameters. We observe that the inference network can often end up sub-optimally sampling very long z (by choosing production rules with many non-terminals), leading to slow and ineffective runs. We therefore cap the run-time to 100 hours—out of ten runs, WW, WS, VIMCO and REINFORCE retain on average 6, 6, 5.75 and 4 runs respectively In Fig. 6.4.1, we show both (i) the quality of the generative model as measured by the average KL between the true and the model production probabilities, and (ii) the quality of the inference network as measured by $\mathbb{E}_{p(x)}[D_{\mathrm{KL}}(p(z|x), q_\phi(z|x))]$ which is estimated up to an additive constant (the conditional entropy $H(p(z|x))$) by the sleep-ϕ loss Eq. (6.5) using samples from the true PCFG.

Quantitatively, WS improves as K increases and outperforms IWAE-based algorithms both in terms of learning and inference amortization. While WW's inference amortization improves slightly as K increases, it is significantly worse than WS's. This is because IS proposals will rarely produce a parse tree z for which $s(z)$ matches x, leading to extremely biased estimates of the wake-ϕ update. In this case, this bias is more significant than that of the data-distribution which can harm the sleep-ϕ update.

We inspect the quality of the inference network by sampling from it. Figure 6.4.2, shows samples from an inference network trained with WS, conditioned on the sentence "astronomers saw stars with telescopes", weighted according to the frequency of occurrence. Section 6.A further includes samples from an inference network trained with VIMCO, showing that none of them match the given sentence ($s(z) \neq x$), and whose production probabilities are poor, unlike with RWS.

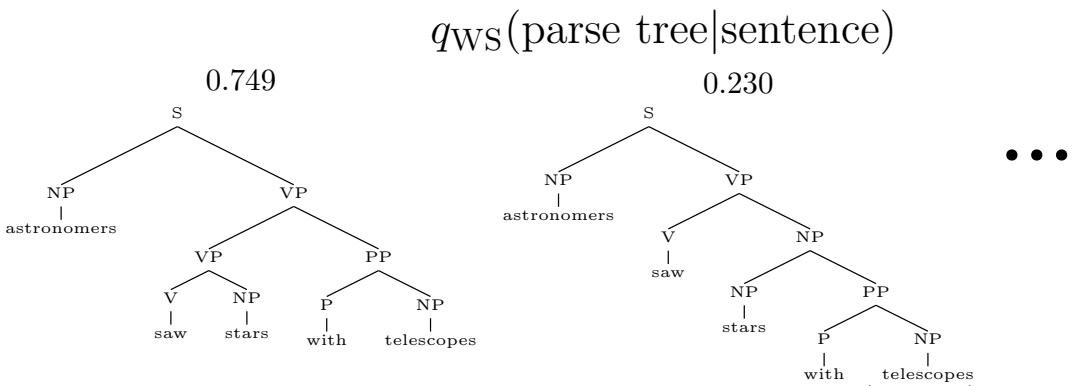


Figure 8.4.2: Samples from the inference network trained with WS ($K = 20$). Highest probability samples correspond to correct sentences ($s(z) = x$).

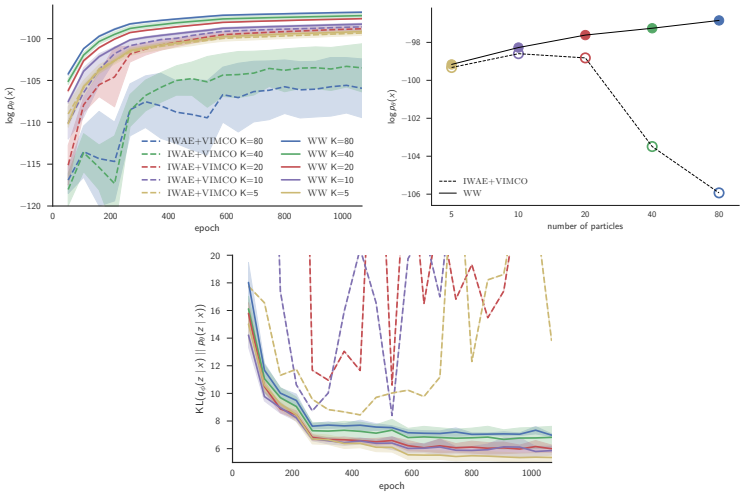


Figure 8.4.3: Training of AIR. *(Left)* Training curves: training with VIMCO leads to larger variance in training than WW. *(Middle)* Log evidence values at the end of training: increasing number of particles improves WW monotonically but improves VIMCO only up to a point ($K = 10$ is the best). *(Right)* WW results in significantly lower variance and better inference networks than VIMCO. Note that KL is between the inference network and the *current* generative model.

8.4.2 Attend, Infer, Repeat

Next, we evaluate WW and VIMCO on AIR (**eslami2016attend**), a structured deep generative model with both discrete and continuous latent variables. AIR uses the discrete variable to decide how many continuous variables are necessary to explain an image. The sequential inference procedure of AIR poses a difficult problem, since it implies a sequential decision process with possible branching. See (**eslami2016attend**) and Section 6.B for the model notation and details.

We set the maximum number of inference steps in AIR to three and train on

50×50 images with zero, one or two MNIST digits. The training and testing data sets consist of 60000 and 10000 images respectively, generated from the respective MNIST train/test datasets. Unlike AIR, which used Gaussian likelihood with fixed standard deviation and continuous inputs (i.e., input $\mathbf{x} \in [0,1]^{50\times 50}$), we use a Bernoulli likelihood and binarized data; the stochastic binarization is similar to **burda2016importance**. Training is performed over two million iterations by RmsProp (**tieleman2012rms**) with the learning rate of 10^{-5}, which is divided by three after 400k and 1000k training iterations. We set the glimpse size to 20×20.

We first evaluate the generative model via the average test log marginal where each log marginal is estimated by a one-sample, 5000-particle IWAE estimate. The inference network is then evaluated via the average test KL from the inference network to the posterior under the current model where each $D_{\mathrm{KL}}(q_\phi(z|x), p_\theta(z|x))$ is estimated as a difference between the log marginal estimate above and a 5000-sample, one-particle IWAE estimate. Note that this estimate is just a proxy to the desired KL from the inference network to the *true* model posterior.

This experiment confirms that increasing number of particles improves VIMCO only up to a point, whereas WW improves monotonically with increased K (Fig. 6.4.3). WW also results in significantly lower variance and better inference networks than VIMCO.

8.4.3 Gaussian Mixture Model

In order to examine the differences between RWS and IWAE more closely, we study a GMM which branches on a discrete latent variable to select cluster assignments. The generative model and inference network are defined as

$$p_\theta(z) = \mathrm{Cat}(z|\mathrm{softmax}(\theta)),\ p(x|z) = \mathcal{N}(x|\mu_z, \sigma_z^2),$$
$$q_\phi(z|x) = \mathrm{Cat}(z|\mathrm{softmax}(\eta_\phi(x))),$$

where $z \in \{0, \ldots, C-1\}$, C is the number of clusters and μ_c, σ_c^2 are fixed to $\mu_c = 10c$ and $\sigma_c^2 = 5^2$. The generative model parameters are $\theta \in \mathbb{R}^C$. The inference network consists of a multilayer perceptron $\eta_\phi : \mathbb{R} \to \mathbb{R}^C$, with the 1-16-$C$ architecture

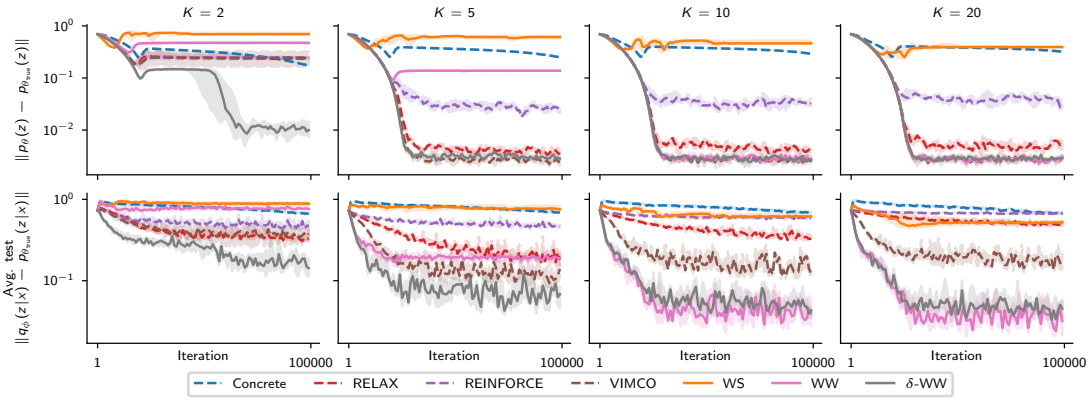


Figure 8.4.4: GMM training. Median and interquartile ranges from 10 repeats shown. *(Top)* Quality of the generative model: WS and WW improve with more particles thanks to lower variance and lower bias estimators of the gradient respectively. IWAE methods suffer with a larger particle budget (**rainforth2018tighter**). WS performs the worst as a consequence of computing the expected KL under the model distribution $p_\theta(x)$ Eq. (6.5) instead of the true data distribution $p(x)$ as with WW Eq. (6.6). WW suffers from branch-pruning (see text) in low-particle regimes, but learns the best model fastest in the many-particle regime; δ-WW additionally learns well in the low-particle regime. *(Bottom)* Both inference network and generative model quality develop identically.

and the tanh nonlinearity, parameterized by ϕ. The chosen family of inference networks is empirically expressive enough to capture the posterior under the true model. The true model is set to $p_{\theta_{\text{true}}}(x)$ where $\text{softmax}(\theta_{\text{true}})_c = (c+5)/\sum_{i=1}^{C}(i+5)$ $(c = 0, \ldots, C-1)$, i.e. the mixture probabilities are linearly increasing with the z . We fix the mixture parameters in order to study the important features of the problem at hand in isolation.

We train using WS, WW, as well as using IWAE with REINFORCE, RELAX, VIMCO and the Concrete distribution. We attempted different variants of relaxations (**rolfe2016dvae**; **vahdat2018dvaepp**) in this setting, but they performed considerably worse than any of the alternatives (c.f. Section 6.E). We fix $C = 20$ and increase number of particles from $K = 2$ to 20. We use the Adam optimizer with default parameters. Each training iteration samples a batch of 100 data points from the true model. Having searched over several temperature schedules for the Concrete distribution, we use the one with the lowest trainable terminal temperature (linearly annealing from 3 to 0.5). We found that using the control variate $c_\rho(g_{1:K}) = \frac{1}{K}\sum_{k=1}^{K} \text{MLP}_\rho([x, g_k])$, with MLP architecture $(1 + C)$-16-16-1

(tanh) led to most stable training (c.f. Section 6.C).

The generative model is evaluated via the L_2 distance between the PMFs of its prior and true prior as $\|\text{softmax}(\theta) - \text{softmax}(\theta_{\text{true}})\|$. The inference network is evaluated via the L_2 distance between PMFs of the current and true posteriors, averaged over a fixed set ($M = 100$) of observations $(x_{\text{test}}^{(m)})_{m=1}^{M}$ from the true model: $\frac{1}{M}\sum_{m=1}^{M} \|q_\phi(z|x_{\text{test}}^{(m)}) - p_{\theta_{\text{true}}}(z|x_{\text{test}}^{(m)})\|$.

We demonstrate that using WS and WW with larger particle budgets leads to better inference networks whereas this is not the case for IWAE methods (Fig. 6.4.4, bottom). Recall that the former is because using more samples to estimate the gradient of the sleep ϕ objective Eq. (6.5) for WS reduces variance at a standard Monte Carlo rate and that using more particles in Eq. (6.7) to estimate the gradient of the wake ϕ objective results in a lower bias. The latter is because using more particles results in the signal-to-noise of IWAE's ϕ gradient estimator to drop at the rate $O(1/\sqrt{K})$ (**rainforth2018tighter**).

Learning of the generative model, through inference-network learning, also monotonically improves with increasing K for WS and WW, but worsens for all IWAE methods except VIMCO, since the θ gradient estimator (common to all methods), $\nabla_\theta \text{ELBO}_{\text{IS}}^{K}(\theta, \phi, x)$ can be seen as an importance sampling estimator whose quality is tied to the proposal distribution (inference network).

To highlight the difference between WW and WS, we study the performance of the generative model and the inference network for different initializations of θ. In Fig. 6.4.4, θ is initialized such that the mixture probabilities are exponentially decreasing with z which results in the data distribution $p_\theta(x)$ being far from $p_{\theta^*}(x)$. Consequently, the sleep-phase ϕ update is highly biased which is supported by WS being worse than WW. On the other hand, if θ is initialized such that the mixture probabilities are equal, $p_\theta(x)$ is closer to $p_{\theta^*}(x)$, which is supported by WS outperforming WW (see Section 6.C.2).

We now describe a failure mode affecting WS, WW, VIMCO, RELAX and REINFORCE due the adverse initialization of θ which we call *branch-pruning*. It is best illustrated by inspecting the generative model and the inference network as

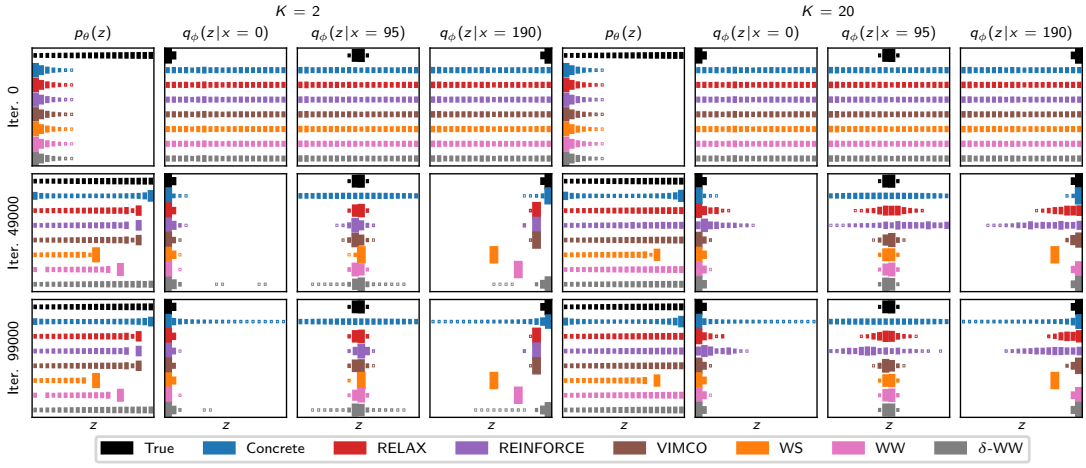


Figure 8.4.5: Generative model and inference network during GMM training shown as Hinton diagrams where areas are proportional to probability. Rows correspond to start, middle and end of optimization. *(Left half)* Learning with few particles leads to the branch-pruning (described in text) of the inference network (shown as conditional PMF given different x) and the generative model (first column of each half) for all methods except δ-WW. Concrete distribution fails. *(Right half)* Learning with many particles leads to branch-pruning only for WS; WW and δ-WW succeed where IWAE fails, learning a suboptimal final generative model.

training progresses, focusing on the low-particle ($K = 2$) regime (Fig. 6.4.5). For WS, the generative model $p_\theta(z)$ peaks at $z = 9$ and puts zero mass for $z > 9$; the inference network $q_\phi(z|x)$ becomes the posterior for this model which, here, has support at most $\{0, \ldots, 9\}$ for all x. This is a local optimum for WS as (i) the inference network already approximates the posterior of the model $p_\theta(z, x)$ well, and (ii) the generative model $p_\theta(z)$, trained using samples from $q_\phi(z|x)$, has no samples outside of its current support. Similar failures occur for WW and VIMCO/RELAX/REINFORCE although the support of the locally optimal $p_\theta(z)$ is larger ($\{0, \ldots, 14\}$ and $\{0, \ldots, 17\}$ respectively).

While this failure mode is a particular feature of the adverse initialization of θ, we hypothesize that WS and WW suffer from it more, as they alternate between two different objectives for optimizing θ and ϕ. WS attempts to amortize inference for the current model distribution $p_\theta(x)$ which reinforces the coupling between the generative model and the inference network, making it easier to get stuck in a local optimum. WW with few particles (say $K = 1$) on the other hand, results in a highly-biased gradient estimator Eq. (6.7) that samples z from $q_\phi(\cdot|x)$ and evaluates $\nabla_\phi \log q_\phi(z|x)$;

this encourages the inference network to concentrate mass. This behavior is not seen in WW with many particles where it is the best algorithm at learning both a good generative model and inference network (Fig. 6.4.4; Fig. 6.4.5, right).

We propose a simple extension of WW, denoted δ-WW, that mitigates this shortcoming by changing the proposal of the self-normalized importance sampling estimator in Eq. (6.7) to $q_{\phi,\delta}(z|x) = (1-\delta)q_\phi(z|x) + \delta\text{Uniform}(z)$. We use $\delta = 0.2$, noting that the method is robust to a range of values. Using a different proposal than the inference network $q_\phi(z|x)$ means that using the low-particle estimator in Eq. (6.7) no longer leads to branch-pruning. This is known as defensive importance sampling (**hesterberg1995weighted**), and is used to better estimate integrands that have long tails using short-tailed proposals. Using δ-WW outperforms all other algorithms in learning both the generative model and the inference network in the low-K regime and performs similarly as WW in the high-K regime.

8.5 Discussion

The central argument here is that where one needs both amortization and model learning for SCFMs, the RWS family of methods is preferable to IWAE with either continuous relaxations or control-variates. The PCFG experiment (Section 6.4.1) demonstrates a setting where continuous relaxations are inapplicable due to potentially infinite recursion, but where RWS applies and WS outperforms all other methods. The AIR experiment (Section 6.4.2) highlights a case where with more particles, performance of VIMCO degrades for the inference network (**rainforth2018tighter**) and consequently the generative model as well, but where RWS's performance on both increases monotonically. Finally, the analysis on GMMs (Section 6.4.3) focuses on a simple model to understand nuances in the performances of different methods. Beyond implications from prior experiments, it indicates that for the few-particle regime, the WW gradient estimator can be biased, leading to poor learning. For this, we design an alternative involving defensive sampling that ameliorates the issue. The precise choice of which variant of RWS to employ depends on which of the two kinds of gradient bias described in Section 6.3.3 dominates. Where the

data distribution bias dominates, as with the AIR experiment, WW is preferable, and where the self-normalized IS bias dominates, as in the PCFG experiment, WS is preferable. In the GMM experiment, we verify this empirically by studying two optimization procedures with low and high data distribution biases.

Acknowledgments

TAL's research leading to these results is supported by EPSRC DTA and Google (project code DF6700) studentships. AK's and YWT's research leading to these results are supported by funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 617071. NS is supported by EPSRC/MURI grant EP/N019474/1. FW's research leading is supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1; DARPA PPAML through the U.S. AFRL under Cooperative Agreement FA8750-14-2-0006; Intel and DARPA D3M, under Cooperative Agreement FA8750-17-2-0093.

Appendix

8.A Probabilistic Context-Free Grammars

We show the *astronomers* PCFG in Fig. 6.A.1. Figure 6.A.2 shows samples from an inference network trained with VIMCO with $K = 20$, conditioned on the sentence $x =$ "astronomers saw stars with telescopes". Figure 6.A.3 shows production probabilities of the non-terminal NP learned by VIMCO and WS with $K = 20$.

$$
\begin{aligned}
\mathrm{S} &\to \mathrm{NP\,VP}\,(1.0) \\
\mathrm{NP} &\to \mathrm{NP\,PP}\,(0.4)|\mathrm{astronomers}\,(0.1)|\mathrm{ears}\,(0.18)| \\
&\quad\ \mathrm{saw}\,(0.04)|\mathrm{stars}\,(0.18)|\mathrm{telescopes}\,(0.1) \\
\mathrm{VP} &\to \mathrm{V\,NP}\,(0.7)|\mathrm{VP\,PP}\,(0.3) \\
\mathrm{PP} &\to \mathrm{P\,NP}\,(1.0) \\
\mathrm{P} &\to \mathrm{with}\,(1.0) \\
\mathrm{V} &\to \mathrm{saw}\,(1.0).
\end{aligned}
$$

Figure 8.A.1: *The astronomers* PCFG from **manning1999foundations**. The terminals are $\{\mathrm{astronomers}, \mathrm{ears}, \mathrm{saw}, \mathrm{stars}, \mathrm{telescopes}, \mathrm{with}\}$, the non-terminals are $\{\mathrm{S}, \mathrm{NP}, \mathrm{VP}, \mathrm{PP}, \mathrm{P}, \mathrm{V}\}$ and the start symbol is S. Each row above lists production rules $\{n_i \to \zeta_j\}$ with the corresponding probabilities p_{ij} in the format $n_i \to \zeta_1\,(p_{i1})|\zeta_2\,(p_{i2})|\cdots|\zeta_J\,(p_{iJ})$.

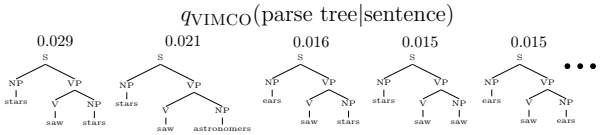


Figure 8.A.2: Samples from the inference network which was trained with VIMCO with $K = 20$.

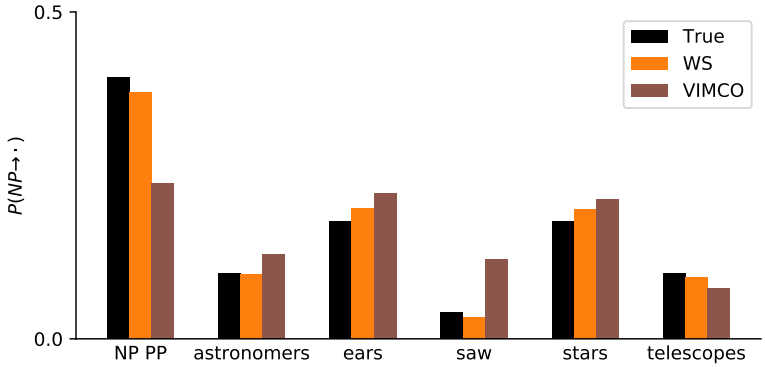


Figure 8.A.3: Production probabilities for the non-terminal NP learned via WS and VIMCO with $K = 20$.

8.B Attend, Infer, Repeat

AIR is a model with many components and might be difficult to understand if not described explicitly. Here, we outline details of our implementation and provide pseudo-code for the inference (Algorithm 4) and generative models (Algorithm 5) in the case of continuous data and Gaussian data likelihood.

Algorithm 4: Inference in AIR

Input : Image $\boldsymbol{x}$,
maximum number of inference steps N

$\boldsymbol{h}_0, \boldsymbol{z}_0^{\text{what}}, \boldsymbol{z}_0^{\text{where}} = \text{initialize}()$
for $n \in [1, \dots, N]$ **do**
  $\boldsymbol{w}_n, \boldsymbol{h}_n = R_\phi\left(\boldsymbol{x}, \boldsymbol{z}_{n-1}^{\text{what}}, \boldsymbol{z}_{n-1}^{\text{where}}, \boldsymbol{h}_{n-1}\right)$
  $p_n \sim \text{Bernoulli}(p \mid \boldsymbol{w}_n)$
  if $p_n = 0$ **then**
    break
  $\boldsymbol{z}_n^{\text{where}} \sim q_\phi^{\text{where}}\left(\boldsymbol{z}^{\text{where}} \mid \boldsymbol{w}_n\right)$
  $\boldsymbol{g}_n = \text{STN}\left(\boldsymbol{x}, \boldsymbol{z}_n^{\text{where}}\right)$
  $\boldsymbol{z}_n^{\text{what}} \sim q_\phi^{\text{what}}\left(\boldsymbol{z}^{\text{what}} \mid \boldsymbol{g}_n\right)$

Output : $\boldsymbol{z}_{1:n}^{\text{what}}$, $\boldsymbol{z}_{1:n}^{\text{where}}$, n

8.C Gaussian Mixture Model

8.C.1 Control Variates

Here, we present the architectures for the REBAR/RELAX control variate used in the GMM experiment.

Algorithm 5: Generation in AIR

Input : $\boldsymbol{z}^{\text{what}}_{1:n}$, $\boldsymbol{z}^{\text{where}}_{1:n}$, n

$\boldsymbol{y}_0 = \mathbf{0}$

for $t \in [1, \ldots, n]$ **do**

  $\hat{\boldsymbol{g}}_t = h^{\text{dec}}_\theta\left(\boldsymbol{z}^{\text{what}}_t\right)$

  $\boldsymbol{y}_t = \boldsymbol{y}_{t-1} + \text{STN}^{-1}\left(\hat{\boldsymbol{g}}_t, \boldsymbol{z}^{\text{where}}_t\right)$

$\hat{\boldsymbol{x}} \sim \text{Normal}\left(\boldsymbol{x} \mid \boldsymbol{y}_n, \sigma^2_x \boldsymbol{I}\right)$

Output : $\hat{\boldsymbol{x}}$

The reparameterized sampling of Gumbels and conditional Gumbels is described by **tucker2017rebar** and **grathwohl2018backpropagation**. In the following, we describe architectures used for the GMM experiment (Section 6.4.3).

REBAR proposes the following architecture for the control variate $c_\rho(g_{1:K})$:

$$c^{\text{REBAR}}_\rho(g_{1:K}) = \rho_1 \log\left(\frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(\text{sm}(g_k/e^{\rho_2}), x)}{q_\phi(\text{sm}(g_k/e^{\rho_2})|x)}\right), \tag{8.8}$$

where $\rho = (\rho_1, \rho_2)$, and sm refers to the softmax function. While the functional form of Eq. (8.8) suggests that it will be highly correlated with $\log(\frac{1}{K}\sum_{k=1}^{K} w_k)$, the terms PMFs in the fraction are undefined due to the softmax. A straightforward fix is to evaluate "a soft PMF" instead:

$$\begin{aligned}
&p_\theta(\text{sm}(g_k/e^{\rho_2}), x)\\
&= p_\theta(\text{sm}(g_k/e^{\rho_2}))p(x|\text{sm}(g_k/e^{\rho_2}))\\
&= \text{Categorical}(\text{sm}(g_k/e^{\rho_2})|\text{sm}(\theta))\cdot\\
&\qquad \text{Normal}(x|\mu_{\text{sm}(g_k/e^{\rho_2})}, \sigma^2_{\text{sm}(g_k/e^{\rho_2})})\\
&\approx \text{sm}(g_k/e^{\rho_2})^\intercal \text{sm}(\theta)\cdot\\
&\qquad \text{Normal}(x|\mu^\intercal \text{sm}(g_k/e^{\rho_2}), (\sigma^2)^\intercal \text{sm}(g_k/e^{\rho_2})),\\
&q_\phi(\text{sm}(g_k/e^{\rho_2})|x)\\
&= \text{Categorical}(\text{sm}(g_k/e^{\rho_2})|\text{sm}(\eta_\phi(x)))\\
&\approx \text{sm}(g_k/e^{\rho_2})^\intercal \text{sm}(\eta_\phi(x)).
\end{aligned}$$

Optimization of the log-temperature ρ_2 is highly sensitive as low values can make the training unstable.

RELAX proposes using an arbitrary neural network for $c_\rho(g_{1:K})$. Due to the symmetry in the arguments, we pick the following for the GMM experiment:

$$c_\rho^{\text{RELAX}}(g_{1:K}) = \frac{1}{K}\sum_{k=1}^{K} \text{MLP}_\rho([x, g_k]), \tag{8.9}$$

where the architecture of the MLP is $(1+C)$-16-16-1 (with the tanh nonlinearity between layers) and ρ are the weights are its weights. This architecture is—unlike the one in Eq. (6.8)—well-defined for all inputs. A drawback of using such control variate is that it can start out not being very correlated with $\log(\frac{1}{K}\sum_{k=1}^{K} w_k)$.

RELAX also proposes using a summation of the free-form control variate like the one in Eq. (6.9) and a more correlated control variate in Eq. (6.8).

We have tried all architectures and found that Eq. (6.9) leads to the most stable and best training.

Using REBAR/RELAX for more complicated models is possible, however designing an architecture that is highly correlated with the high-variance term and stable to train still remains a challenge.

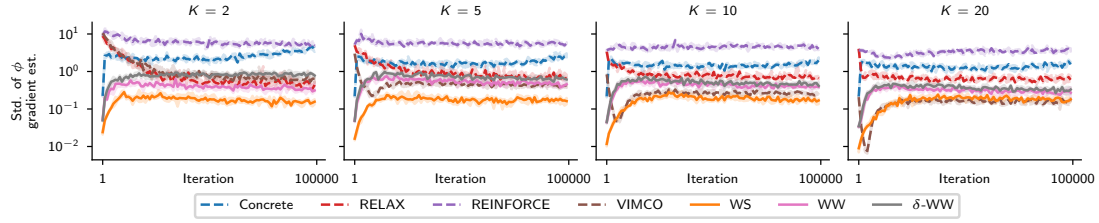


Figure 8.C.1: Standard deviation of gradient estimator of ϕ for GMM. Median and interquartile ranges from 10 repeats shown. WW and WS have lower-variance gradient estimators of ϕ than IWAE except VIMCO, as they avoid the high-variance term ① in (6.2). This is a necessary, but not sufficient, condition for efficient learning, with other factors being gradient direction and the ability to escape local optima. The standard deviation of ϕ's gradient estimator is given by $\frac{1}{D_\phi}\sum_{d=1}^{D_\phi} \text{std}(g_d)$ where g_d is the dth (out of D_ϕ) element of one of ϕ's gradient estimators (e.g. Eq. (6.2) for REINFORCE) and std$(\cdot)$ is estimated using 10 samples.

8.C.2 Additional Results

Here, we include additional GMM experiments: one for studying ϕ's gradient variance (Fig. 6.C.1), the other for comparing performances of the generative

model and inference networks when θ is initialized closer to θ^* than in the main paper (Fig. 6.C.2).

WW and WS have lower variance gradient estimators than IWAE, except VIMCO. This is because ϕ's gradient estimators for WW and WS do not include the high-variance term ① in Eq. (6.2). This is a necessary but not sufficient condition for efficient learning with other important factors being gradient direction and the ability to escape local optima. Employing the Concrete distribution gives low-variance gradients for ϕ to begin with, but the model learns poorly due to the high gradients bias (due to high temperature hyperparameter).

In Fig. 6.C.2, we initialize θ so that the mixture probabilities are constant. This means that the data bias is smaller than in the main paper's setting. With smaller data bias, we expect WS to perform better. This is empirically verified since WS outperforms other methods, including WW.

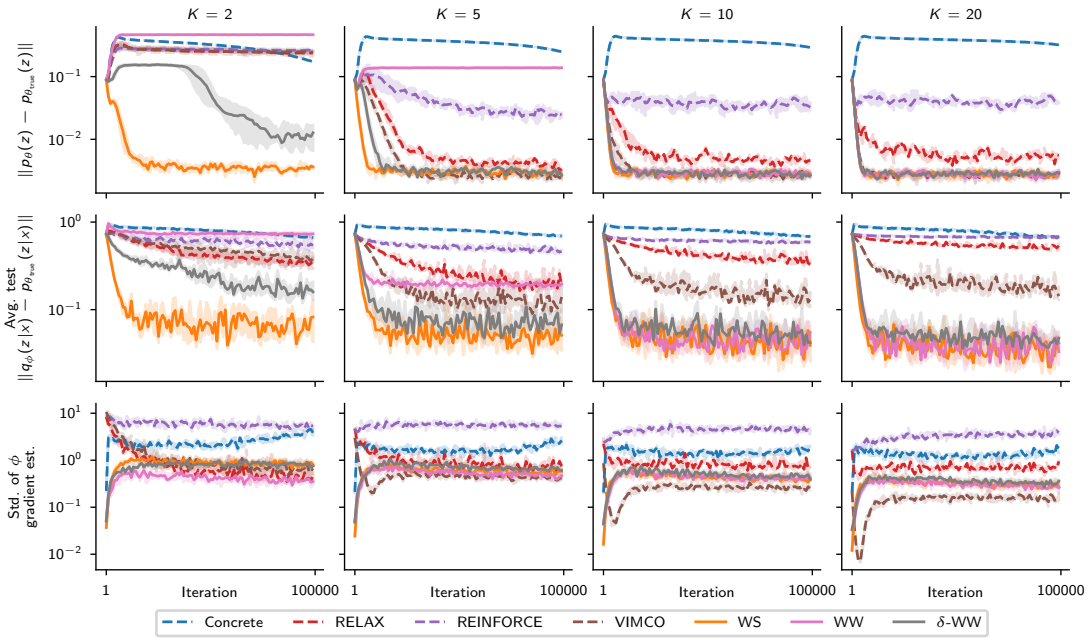


Figure 8.C.2: GMM training when $p_\theta(x)$ is close to $p_{\theta^*}(x)$. WS outperforms other methods including WW in generative model (top) and inference network (middle) learning. VIMCO has the lowest gradient variance (bottom) but still performs worse than WS and results in worsening of the inference network as number of particles is increased.

8.D Sigmoid Belief Networks

In Fig. 6.D.1, we show training of sigmoid belief networks with three stochastic layers with the same architecture as in **mnih2016variational**. We additionally drive number of particles up to $K = 5000$ and include KL plots. We find that in high particle regimes, model learning is virtually the same for WW and VIMCO. However, WW outperforms VIMCO in terms of inference network learning.

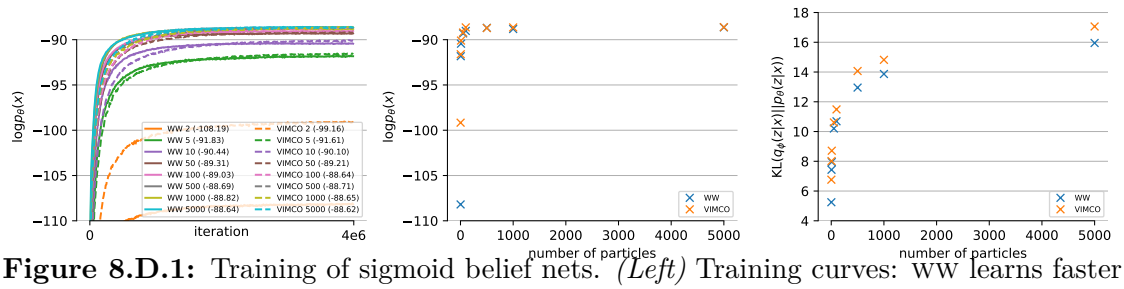


Figure 8.D.1: Training of sigmoid belief nets. *(Left)* Training curves: WW learns faster than VIMCO but results in equal or slightly worse end test log likelihood. *(Middle)* Log evidence values at the end of training: VIMCO is slightly better than WW in low-particle regimes but virtually the same in high-particle regimes. *(Right)* KL divergence at the end of training: WW results in much lower KL divergence than VIMCO.

8.E Discrete VAEs

Rolfe 2016 (**rolfe2016dvae**) introduces discrete VAE (DVAE). It combines a prior over binary latent variables with an element-wise spike-and-X smoothing transformation, allowing approximate marginalization of the discrete variables. This results in a continuous relaxation of discrete variables and a low-variance gradient estimator. **vahdat2018dvaepp** replaced the original transformation with an overlapping exponential transformation, leading to a yet lower-variance gradient estimator. While both approaches produce relaxed binary variables, the relaxation is significantly less tight (**vahdat2018dvaepp**, Appendix C, Figure 5.) then the CONCRETE of **jang2017categorical**; **maddison2017concrete** Both approaches require analytical inverse CDFs of the smoothing transformations, a shortcoming addressed by **vahdat2018dvaehash** — it also leads to a tighter relaxation than its predecessors, however no comparison to CONCRETE is available.

DVAE was designed for undirected binary priors, *e.g.* restricted Boltzmann machines (RBM), and it does not account for the case of categorical latent variables. It is possible to construct a d-dimensional categorical variable from $d-1$ binary variables via stick-breaking construction. This process is slow, however, as it requires $\mathcal{O}(d)$ sequential operations and cannot be parallelized. Moreover, in the case of relaxed variables, the tightness of the derived relaxed categorical variable decreases exponentially with the number of dimensions. This is a major issue in control flows: not only we have to evaluate all branches of the control flow, but the indicator variables that we multiply with outcomes of different branches become exponentially loose with the depth of the flow.

Appendices

Cor animalium, fundamentum eft vitæ, princeps omnium, Microcofmi Sol, a quo omnis vegetatio dependet, vigor omnis & robur emanat.

The heart of animals is the foundation of their life, the sovereign of everything within them, the sun of their microcosm, that upon which all growth depends, from which all power proceeds.

— William Harvey **harvey_exercitatio_1628**

A

Review of Cardiac Physiology and Electrophysiology

Contents

Appendices are just like chapters. Their sections and subsections get numbered and included in the table of contents; figures and equations and tables added up, etc. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et dui sem. Aliquam dictum et ante ut semper. Donec sollicitudin sed quam at aliquet. Sed maximus diam elementum justo auctor, eget volutpat elit eleifend. Curabitur hendrerit ligula in erat feugiat, at rutrum risus suscipit. Pellentesque habitant

morbi tristique senectus et netus et malesuada fames ac turpis egestas. Integer risus nulla, facilisis eget lacinia a, pretium mattis metus. Vestibulum aliquam varius ligula nec consectetur. Maecenas ac ipsum odio. Cras ac elit consequat, eleifend ipsum sodales, euismod nunc. Nam vitae tempor enim, sit amet eleifend nisi. Etiam at erat vel neque consequat.

A.1 Anatomy

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec accumsan cursus neque. Pellentesque eget tempor turpis, quis malesuada dui. Proin egestas, sapien sit amet feugiat vulputate, nunc nibh mollis nunc, nec auctor turpis purus sed metus. Aenean consequat leo congue volutpat euismod. Vestibulum et vulputate nisl, at ultrices ligula. Cras pulvinar lacinia ipsum at bibendum. In ac augue ut ante mollis molestie in a arcu.

Etiam vitae quam sollicitudin, luctus tortor eu, efficitur nunc. Vestibulum maximus, ante quis consequat sagittis, augue velit luctus odio, in scelerisque arcu magna id diam. Proin et mauris congue magna auctor pretium id sit amet felis. Maecenas sit amet lorem ipsum. Proin a risus diam. Integer tempus eget est condimentum faucibus. Suspendisse sem metus, consequat vel ante eget, porttitor maximus dui. Nunc dapibus tincidunt enim, non aliquam diam vehicula sed. Proin vel felis ut quam porta tempor. Vestibulum elit mi, dictum eget augue non, volutpat imperdiet eros. Praesent ac egestas neque, et vehicula felis.

Pellentesque malesuada volutpat justo, id eleifend leo pharetra at. Pellentesque feugiat rutrum lobortis. Curabitur hendrerit erat porta massa tincidunt rutrum. Donec tincidunt facilisis luctus. Aliquam dapibus sodales consectetur. Suspendisse lacinia, ipsum sit amet elementum fermentum, nulla urna mattis erat, eu porta metus ipsum vel purus. Fusce eget sem nisl. Pellentesque dapibus, urna vitae tristique aliquam, purus leo gravida nunc, id faucibus ipsum magna aliquet ligula. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin sem lacus, rutrum eget efficitur sed, aliquam vel augue. Aliquam ut eros vitae sem cursus ultrices ut ornare urna. Nullam tempor porta enim, in pellentesque arcu commodo quis.

Interdum et malesuada fames ac ante ipsum primis in faucibus. Curabitur maximus orci purus, ut molestie turpis pellentesque ut.

Donec lacinia tristique ultricies. Proin dignissim risus ut dolor pulvinar mollis. Proin ac turpis vitae nibh finibus ullamcorper viverra quis felis. Mauris pellentesque neque diam, id feugiat diam vestibulum vitae. In suscipit dui eu libero ultrices, et sagittis nunc blandit. Aliquam at aliquet ex. Nullam molestie pulvinar ex vitae interdum. Praesent purus nunc, gravida id est consectetur, convallis elementum nulla. Praesent ex dolor, maximus eu facilisis at, viverra eget nulla. Donec ullamcorper ante nisi. Sed volutpat diam eros. Nullam egestas neque non tortor aliquet, sed pretium velit tincidunt. Aenean condimentum, est ac vestibulum mattis, quam augue congue augue, mattis ultrices nibh libero non ante. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Aenean volutpat eros tortor, non convallis sapien blandit et. Maecenas faucibus nulla a magna posuere commodo. Nullam laoreet ante a turpis laoreet malesuada. Phasellus in varius sem. Vestibulum sagittis nibh sed tincidunt blandit. Donec aliquam accumsan odio sit amet lacinia. Integer in tellus diam. Vivamus varius massa leo, vitae ullamcorper metus pulvinar sed. Maecenas nec lorem ornare, elementum est quis, gravida massa. Suspendisse volutpat odio ex, ac ultrices leo ultrices vel. Sed sed convallis ipsum. Pellentesque euismod a nulla sed rhoncus. Sed vehicula urna vitae mi aliquet, non sodales lacus ullamcorper. Duis mattis justo turpis, id tempus est tempus eu. Curabitur vitae hendrerit ligula.

Curabitur non pretium enim, in commodo ligula. Etiam commodo eget ligula a lacinia. Vestibulum laoreet ante tellus, vel congue sapien ornare in. Donec venenatis cursus velit vitae pulvinar. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Suspendisse in metus lectus. Pellentesque gravida dolor eget finibus imperdiet. Duis id molestie tortor. Mauris laoreet faucibus facilisis. Aliquam vitae dictum massa, sit amet dignissim lacus.

Fusce eleifend tellus id ex consequat maximus. Donec ultrices ex ut turpis ornare, non molestie mi placerat. Nulla sit amet auctor nunc, sit amet euismod elit. Phasellus risus tellus, condimentum a metus et, venenatis tristique urna. Cras mattis

felis eget ipsum fermentum egestas. Ut augue odio, venenatis id convallis vel, congue quis augue. Maecenas sed maximus est, posuere aliquet tortor. Ut condimentum egestas nisi eu porttitor. Ut mi turpis, posuere id lorem vel, elementum tempor arcu.

Morbi nisl arcu, venenatis non metus ac, ullamcorper scelerisque justo. Nulla et accumsan lorem. Mauris aliquet dui sit amet libero aliquet, in ornare metus porttitor. Integer ultricies urna eu consequat ultrices. Maecenas a justo id purus ultricies posuere sed et quam. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Sed eleifend risus quis aliquet gravida. Nullam ac erat porta est bibendum dictum in a dolor. Nam eget turpis viverra, vulputate lectus eget, mattis ligula. Nam at tellus eget dui lobortis sodales et ut augue. In vestibulum diam eget mi cursus, ut tincidunt nulla pellentesque.

Aliquam erat volutpat. Sed ultrices massa id ex mattis bibendum. Nunc augue magna, ornare at aliquet gravida, vehicula sed lorem. Quisque lobortis ipsum eu posuere eleifend. Duis bibendum cursus viverra. Nam venenatis elit leo, vitae feugiat quam aliquet sed. Cras velit est, tempus ac lorem sed, pharetra lobortis ipsum. Donec suscipit gravida interdum. Nunc non finibus est. Nullam turpis elit, tempus non ante.

A.2 Mechanical Cycle

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean tellus est, suscipit sed facilisis quis, malesuada at ipsum. Nam tristique urna quis quam iaculis, et mattis orci pretium. Praesent euismod elit vel metus commodo ultrices. Vestibulum et tincidunt ex, in molestie ex. Donec ullamcorper sollicitudin accumsan. Etiam ac leo turpis. Duis a tortor felis. Nullam sollicitudin eu purus ac hendrerit. Nam hendrerit ligula libero, eget finibus orci bibendum a. Aenean ut ipsum magna.

Ut viverra, sapien sed accumsan blandit, nisi sem tempus tellus, at suscipit magna erat ornare nunc. Proin lacinia, nisi ut rutrum malesuada, nibh quam pellentesque nunc, sit amet consectetur purus felis ac tortor. Suspendisse lacinia ipsum eu sapien pellentesque mattis. Mauris ipsum nunc, placerat non diam vel, efficitur laoreet nunc. Sed lobortis, ipsum eget gravida facilisis, sem nulla viverra

mi, in placerat eros sem viverra lacus. Aliquam porta aliquet diam vel commodo. Nulla facilisi. Duis erat libero, lobortis vel hendrerit vitae, sagittis id dui. Nulla pretium eros nec quam tincidunt, vel luctus mi aliquam. Integer imperdiet purus in est tristique venenatis. Ut pellentesque, nunc vitae iaculis ultricies, urna turpis dignissim risus, a laoreet felis magna nec erat.

Quisque sollicitudin faucibus ligula, et egestas nibh dictum sit amet. Proin eu mi a lectus congue pretium eu quis arcu. Suspendisse vehicula libero eu ipsum aliquam, vel elementum nibh mattis. Sed sed sapien vitae turpis tristique pulvinar a ut metus. Etiam semper gravida est, mollis gravida est porta ac. Proin eget tincidunt erat. Maecenas ultrices erat eget purus ultricies, ut lacinia arcu dictum. Nam et nisi sit amet ex congue mattis vel eget lorem. Aliquam erat volutpat. Pellentesque porttitor nibh vitae elementum consectetur. Aenean et est lobortis, congue sapien non, ullamcorper sapien. Ut facilisis sem non dapibus vehicula.

Mauris euismod odio dolor, sit amet gravida mauris placerat et. Curabitur nec dolor non nibh molestie lobortis dignissim non ante. Nullam rutrum lobortis ultrices. Aenean ex erat, elementum sed maximus id, posuere id quam. Proin rutrum ex elit, pretium aliquam risus finibus at. Aenean egestas orci velit, sed aliquet sapien condimentum a. Duis consequat, arcu eu viverra venenatis, dolor lorem gravida lectus, non aliquet nisi sem at augue. Donec laoreet blandit luctus. Aenean vehicula nisl vel faucibus luctus. Sed ut semper velit, vitae laoreet magna. Sed at interdum magna.

Sed iaculis faucibus odio, eu aliquam purus efficitur vel. Cras at nulla ac enim congue varius ut et nulla. Integer blandit mattis augue.

A.3 Electrical Cycle

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In faucibus condimentum rhoncus. Ut dictum nisl id risus gravida lobortis. Sed vehicula mollis tellus ut varius. Fusce eget egestas dui, et commodo dui. Proin sollicitudin interdum tempus. Nullam in elit a enim fringilla bibendum. Vestibulum sodales pellentesque condimentum. Nulla facilisi. Nunc et dolor in nulla eleifend dictum at vel ligula. Aliquam ut velit

non elit ullamcorper porta ac et ex. Fusce ornare magna non nunc vestibulum, eget molestie quam dictum. In interdum aliquam odio, in posuere tellus convallis quis. Curabitur non diam elit. Proin vulputate orci diam, a tincidunt ante luctus eu. Ut a viverra ligula. Curabitur pulvinar tempus tellus eget suscipit.

Aliquam posuere massa at ante dapibus congue. Curabitur ullamcorper tortor eget consectetur aliquet. Mauris tempor magna id mauris fringilla, a varius erat blandit. Nam eleifend ullamcorper placerat. Phasellus augue tortor, volutpat bibendum lorem nec, fringilla volutpat nisl. Mauris cursus urna metus, vel eleifend orci iaculis ut. Sed sit amet scelerisque massa, quis consequat dui. Donec semper sem dui, ac placerat velit egestas vel. Nulla facilisi. Quisque tellus eros, sagittis malesuada augue ut, faucibus dictum nulla. Vestibulum non dapibus erat, ut consequat libero. Ut turpis mi, dapibus commodo libero lobortis, maximus vestibulum lectus. Vestibulum sit amet sapien dapibus, tincidunt leo in, suscipit arcu. Sed in erat bibendum, laoreet eros eu, pellentesque justo. Nulla sodales purus neque, eget maximus ipsum consequat at. Maecenas a nisl sagittis, tempus ipsum sed, dictum mauris.

Suspendisse posuere odio lacus, at auctor tortor vehicula sed. Phasellus suscipit ornare enim vitae placerat. Sed viverra purus vel sapien tempor, quis iaculis erat laoreet. Aenean vel nunc vestibulum, ornare nunc ac, mollis urna. Aenean ultrices felis ipsum, ac semper est ullamcorper in. Donec in justo varius, egestas tortor ut, venenatis augue. Duis mattis, ligula quis lacinia fringilla, tellus neque accumsan ipsum, vitae tempor metus elit vel nibh. Curabitur porttitor urna nec sapien tempor, et porttitor velit malesuada.

Suspendisse aliquam nisl quis placerat vulputate. Proin dapibus ipsum ac ante sagittis, volutpat auctor sem dapibus. Nam in facilisis odio. Integer ante mauris, eleifend et pulvinar in, venenatis quis ligula. Phasellus posuere sollicitudin tortor eget euismod. Maecenas mollis tortor eget justo vulputate sagittis. Etiam hendrerit massa quis ex molestie sodales. Quisque facilisis erat lacus, id convallis sem suscipit bibendum. Integer dui urna, pharetra sed porta sed, bibendum ut odio. Donec placerat at lectus egestas consequat. Sed id rhoncus est, vitae vulputate sapien.

Fusce tempus quam lorem, id ornare turpis sodales sed. Integer aliquet urna eget condimentum consequat. Vestibulum quis dui vel ligula posuere luctus id nec turpis.

Nam vitae placerat lacus. Mauris scelerisque interdum volutpat. Nunc aliquet tristique enim, sit amet molestie felis ullamcorper vitae. Nullam sollicitudin orci orci, in condimentum tellus consectetur in. Nam id justo justo. Fusce eget finibus est. Proin id tortor nec quam cursus vehicula. Aliquam ultrices eros eros, a tincidunt elit eleifend auctor.

Nullam consectetur dapibus ligula sit amet efficitur. Nunc non posuere sapien. Vivamus dui nisl, aliquam id ipsum non, pulvinar ornare neque. Nunc rhoncus pretium congue. Fusce id laoreet enim. Cras sed massa in eros bibendum auctor in nec sem. Nam commodo, velit id porta consequat, mi arcu gravida lorem, ut aliquam elit ante quis dui. Quisque in massa sed nibh blandit dictum.

Vestibulum molestie consectetur porttitor. Donec tincidunt vel orci at pharetra. Nullam id felis sit amet nulla tempus lacinia. Integer egestas ullamcorper massa, ut ultricies diam congue sit amet. Cras sit amet velit at nibh vehicula finibus a et lorem. Cras odio metus, venenatis ut ultrices non, ornare ac orci. Morbi et nulla dui. Mauris dictum molestie nibh, eu efficitur lorem accumsan quis.

A.4 Cellular Electromechanical Coupling

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam vitae consectetur metus, ac maximus ex. Quisque vitae ex eu lectus ultricies consequat vel non lorem. Etiam odio ipsum, tempus ut lobortis in, molestie ac leo. Vivamus mollis feugiat bibendum. Vestibulum eget venenatis quam. Aenean faucibus, massa sed ullamcorper porta, arcu nunc iaculis velit, quis consectetur purus neque placerat nibh. Vestibulum elit nunc, dignissim vulputate venenatis et, sodales non massa. Proin leo ligula, vehicula eu aliquam varius, posuere a dolor. Donec iaculis auctor neque, sit amet gravida libero porta vel. Vivamus consequat elementum lacus, at bibendum mauris egestas nec. Fusce fermentum diam eu dolor ornare, vitae vestibulum leo interdum. Morbi luctus libero quis dictum laoreet. Etiam semper porta ante, vel ullamcorper enim sodales quis.

Nullam eu nisi faucibus, fermentum ex auctor, tempor arcu. Phasellus condimentum erat mi, condimentum malesuada ligula congue venenatis. Nullam gravida imperdiet urna quis cursus. Ut tempus nec purus eget posuere. Cras non nulla sit amet justo aliquet pellentesque nec sed eros. Nam aliquam nisl urna, in placerat magna gravida venenatis. Donec interdum vel magna ullamcorper molestie. Nunc felis neque, rhoncus fringilla faucibus sit amet, ultrices sed magna. Maecenas malesuada hendrerit diam in ultrices. Nam libero urna, volutpat ut auctor eget, interdum sed odio. Vestibulum suscipit mauris nec augue ornare, ut eleifend nulla gravida. Proin imperdiet, mauris quis consectetur porta, leo dui convallis leo, id lobortis massa diam eu libero. Aenean hendrerit vel ante aliquam venenatis. Pellentesque bibendum pretium odio, ut sagittis lectus feugiat a. Donec porttitor vulputate lacus.

Nunc volutpat efficitur lacus in aliquet. Nullam non iaculis diam, at ultrices diam. Proin vehicula vulputate cursus. Morbi tempus sapien id urna lobortis interdum. Maecenas elementum sagittis elementum. Donec at sodales velit, a posuere tortor. Nulla id hendrerit tortor. Sed semper velit in magna sagittis pulvinar. Nulla nec arcu molestie, ultricies sapien sit amet, sollicitudin nisi. Donec nisi massa, suscipit ut dignissim quis, lacinia id leo.

Suspendisse ut mi metus. Morbi tincidunt ligula in porttitor consectetur. Integer eu urna urna. Suspendisse potenti. Mauris sit amet felis eu diam auctor ullamcorper. Morbi in porta nisi. Nam ante tortor, venenatis vitae tempor sed, sagittis vitae velit. In semper orci sit amet nisi ullamcorper varius. Aenean dignissim ultrices imperdiet. Maecenas lacinia enim id neque porttitor iaculis. Curabitur laoreet ante ut urna dignissim, id sollicitudin metus consectetur. Aenean massa ipsum, auctor vel ante vel, blandit dignissim libero. Fusce interdum ac magna et interdum.