Near-Optimal Glimpse Sequences for Improved Hard Attention Neural Network Training

William Harvey
Department of Computer Science
University of British Columbia
Vancouver, Canada
wsgh@cs.ubc.ca

Michael Teng
Department of Engineering Science
University of Oxford
Oxford, United Kingdom
mteng@robots.ox.ac.uk

Frank Wood
Department of Computer Science
University of British Columbia
Vancouver, Canada
fwood@cs.ubc.ca

Abstract

We introduce the use of Bayesian optimal experimental design techniques for generating glimpse sequences to use in semi-supervised training of hard attention networks. Hard attention holds the promise of greater energy efficiency and superior inference performance. Employing such networks for image classification usually involves choosing a sequence of glimpse locations from a stochastic policy. As the outputs of observations are typically non-differentiable with respect to their glimpse locations, unsupervised gradient learning of such a policy requires REINFORCE-style updates. Also, the only reward signal is the final classification accuracy. For these reasons hard attention networks, despite their promise, have not achieved the wide adoption that soft attention networks have and, in many practical settings, are difficult to train. We find that our method for semi-supervised training makes it easier and faster to train hard attention networks and correspondingly could make them practical to consider in situations where they were not before.

1 Introduction

A reasonable definition of attention is the "allocation of limited cognitive processing resources" [1]. In humans the density of photoreceptors varies across the retina, is much greater in the fovea [4], and we have an approximately 210 degree field of view [26]. Taken together these facts mean that the visual system is a limited resource with respect to observing the environment and that it must be allocated, or controlled, by some attention mechanism. We refer to this kind of controlled allocation of limited sensor resources as "hard" attention. We call another kind of attention, the controlled application of limited computational resources to full sensory input as "soft" attention. Our focus is on hard attention mechanisms in this paper.

Hard visual attention mechanisms have some advantages, foremost amongst them is that it is possible that they can solve tasks using orders of magnitude less computation and sensor bandwidth than alternatives [23]. This alone is reason enough to try to design artificial systems that make use of hard attention [3, 12, 20, 31]. Also, as the physiologically inspired design of convolutional neural network architectures yielded implicit regularization that was beneficial in terms of application performance, one could reasonably expect the same to be true of hard attention mechanisms. Put another way;

Preprint. Under review.

since we perceive the world using a hard attention mechanism, neural networks programmed with similar computational structure should be more likely to form internal representations like ours.

A disadvantage of most artificial hard visual attention mechanisms is that they require choosing a sequence of attention locations [20], with respect to which their output is non-differentiable. Soft visual attention mechanisms [28], on the other hand, compute a weighted average of embeddings taken at all possible attention locations in an image. This means that the outputs of soft attention mechanisms are differentiable with respect to their parameters and thus are amenable to training via standard gradient backpropagation techniques. Training hard attention mechanisms is a reinforcement learning task equivalent to policy learning where glimpse locations are actions and task success, classification accuracy in this paper, is the reward. High variance gradient estimation techniques such as REINFORCE [30] must be used, and this causes learning problems, particularly when the required glimpse sequences are long. Long glimpse sequences are required when either or both the task is hard or the amount of information gathered in each glimpse region is small. For this reason there is a trade-off between the efficiency of hard attention network training and the efficiencies that can be gained by making the per-glimpse information, and thereby computation, smaller.

One of the most studied aspects of hard visual attention is trying to explain where one looks when solving a task. A coherent account of this now appears throughout neuroscience, namely, that visual attention is directed so as to maximally reduce entropy in an agent's world model [5, 9, 14, 24]. There is a corresponding machine learning literature which provides a general mathematical formulation of such an objective, namely Bayesian experimental design [6, 10].

What we explore in this paper is whether hard attention neural networks with known architectures for image classification can be trained more efficiently if given semi-supervision by annotating a subset of their training examples with glimpse locations that are near-optimal in the Bayesian experimental design sense. We find that providing even a very small amount of supervision in the form of these glimpse location paths helps training tremendously.

The majority of this paper consists of an account of the various algorithmic innovations we made in order to generate such a supervision signal. Amongst them we invented a new form of structured dropout that corresponds to focusing attention on a small portion of an image and developed a patch-based image search procedure that enables conditional image completion using a non-parametric image distribution. We show how to use these tools within a Bayesian optimal experimental design framework to generate fully-observed glimpse location sequences and then demonstrate the positive effects of including those when training hard attention neural networks to classify various attributes of faces.

2 Background

Throughout this paper, we consider using attention for image classification: given an image, $\mathbf{x}^{(\mathbf{i})}$, we attempt to infer its kth attribute, $\theta_k^{(i)}$. The different attributes of an image correspond to different possible classification tasks. For notational clarity, we shall from now on refer to the image as $\mathbf{x}$, and attribute label as θ, with the task and image index implicit. We now review hard attention mechanisms and Bayesian experimental design.

Hard Attention We consider a model of hard visual attention consisting of a recurrent neural network which outputs a distribution over a *location*, $\mathbf{l}_t$, at each time step. This describes which part of the input image to attend to at the next time step, t. A deterministic *glimpse sensor* then extracts a *glimpse*, $\mathbf{y}_t$, of the image at this location and feeds it to the neural network, along with an embedding of $\mathbf{l}_t$. A glimpse in our model consists of a fixed-size square of contiguous pixels. At some final time, T, the neural network produces an output, which is the classification prediction in our case. A loss is calculated using this output, and used to optimise all network parameters. Throughout this paper, T is treated as a constant. This structure corresponds most closely to that of Mnih et al. [20], although they use a glimpse consisting of a multi-resolution foveated view. Other variations in the literature include networks which produce outputs at multiple time steps or have a varying sequence length, particularly for text generation and multiple object recognition [3, 31].

Typically, it is not possible to differentiate the output of the glimpse sensor with respect to the location. This means that the attention mechanism cannot be trained by standard backpropagation of the gradient. Instead, it is trained using a REINFORCE [30] estimate of the gradient of the

expectation of the loss over all possible glimpse sequences. Let $\mathcal{L}^i_\phi(\mathbf{l}_{1:T})$ denote the deterministic classification loss on training example i with given network parameters ϕ and glimpse sequence $\mathbf{l}_{1:T}$, and $p_\phi(\mathbf{l}_{1:T}|\mathbf{x}^i)$ denote the policy (i.e., the distribution over attention sequences on image $\mathbf{x}^i$ defined by the network parameters). Then the gradient of $\mathbb{E}_{p_\phi(\mathbf{l}_{1:T}|\mathbf{x}^i)}\left[\mathcal{L}^i_\phi(\mathbf{l}_{1:T})\right]$, the expectation of this loss over the policy, is given by

$$\frac{\partial}{\partial\phi}\mathcal{L}^i_\phi(\mathbf{l}_{1:T}) = \mathbb{E}_{p_\phi(\mathbf{l}_{1:T}|\mathbf{x}^i)}\left[\frac{\partial}{\partial\phi}\mathcal{L}^i_\phi(\mathbf{l}_{1:T}) + \mathcal{L}^i_\phi(\mathbf{l}_{1:T})\frac{\partial}{\partial\phi}\log p_\phi(\mathbf{l}_{1:T}|\mathbf{x}^i)\right]. \tag{1}$$

This expectation lends itself to an approximation by Monte Carlo sampling from the policy, $p_\phi(\mathbf{l}_{1:T}|\mathbf{x}^i)$. Although this gives an unbiased estimate, it has high variance which makes optimisation difficult. Intuitively, the high variance is a result of the estimator having access to just the value of $\mathcal{L}^i_\phi(\mathbf{l}_{1:T})$, and not its derivative with respect to $\mathbf{l}_{1:T}$ [27]. The resulting difficulties only increase as the sequence length and dimensionality of $\mathbf{l}_{1:T}$ increase.

Bayesian Experimental Design Designing an experiment to be maximally informative is a fundamental problem that applies as much to tuning the parameters of a political survey [29] as to deciding where to direct your gaze to answer a query. Bayesian experimental design [6] provides a unifying framework for this by allowing a formal comparison of possible experiments using problem-specific prior knowledge. Consider designing an experiment to infer some parameter, θ. To clarify the connection to attention, we denote the design of the experiment as $\mathbf{l}$. The experiment results in a measurement of $\mathbf{y} \sim p(\mathbf{y}|\mathbf{l},\theta)$, from which a posterior distribution over θ can be inferred. The design should be chosen to maximally reduce our uncertainty in θ. To quantify this, we use the Shannon information gain, which is equivalent to the reduction in entropy between the prior and posterior:

$$\begin{aligned}\mathrm{IG}(\mathbf{y},\mathbf{l}) &= \mathcal{H}\left[p(\theta)\right] - \mathcal{H}\left[p(\theta|\mathbf{y},\mathbf{l})\right] &(2)\\ &= \mathbb{E}_{p(\theta|\mathbf{y},\mathbf{l})}\left[\log p(\theta|\mathbf{y},\mathbf{l})\right] - \mathbb{E}_{p(\theta)}\left[\log p(\theta)\right]. &(3)\end{aligned}$$

While designing the experiment, $\mathbf{y}$ is unknown and so $\mathrm{IG}(\mathbf{y},\mathbf{l})$ cannot be calculated. Instead, we define the expected information gain, $\mathrm{EIG}(\mathbf{l})$, as the expectation of $\mathrm{IG}(\mathbf{y},\mathbf{l})$ over the marginal distribution of $\mathbf{y}$ given $\mathbf{l}$: $p(\mathbf{y}|\mathbf{l}) = \int p(\mathbf{y}|\mathbf{l},\theta)p(\theta)\mathrm{d}\theta$. This is used as a utility function when optimizing the design:

$$\begin{aligned}\mathrm{EIG}(\mathbf{l}) &= \mathbb{E}_{p(\mathbf{y}|\mathbf{l})}\left[\mathcal{H}\left[p(\theta)\right] - \mathcal{H}\left[p(\theta|\mathbf{y},\mathbf{l})\right]\right] &(4)\\ &= \mathcal{H}\left[p(\theta)\right] - \mathbb{E}_{p(\mathbf{y}|\mathbf{l})}\left[\mathcal{H}\left[p(\theta|\mathbf{y},\mathbf{l})\right]\right]. &(5)\end{aligned}$$

Since the prior entropy, $\mathcal{H}\left[p(\theta)\right]$, does not depend on $\mathbf{l}$, designing an experiment to maximize the expected information gain can be equivalently framed as minimizing the expected posterior entropy, $\mathrm{EPE}(\mathbf{l})$, where:

$$\mathrm{EPE}(\mathbf{l}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{l})}\left[\mathcal{H}\left[p(\theta|\mathbf{y},\mathbf{l})\right]\right]. \tag{6}$$

Intuitively, this involves selecting $\mathbf{l}$ in order to minimize the expected uncertainty in the posterior. The above generalizes to sequential experimental design when the prior, $p(\theta)$, is replaced with a posterior conditioned on previous designs and observations, $p(\theta|\mathbf{y}_{1:t-1},\mathbf{l}_{1:t-1})$. The marginal distribution over the new measurement, $\mathbf{y}_t$ is then given by $\int p(\mathbf{y}_t|\theta)p(\theta|\mathbf{y}_{1:t-1},\mathbf{l}_{1:t-1})\mathrm{d}\theta$, leading to the objective:

$$\mathrm{EPE}_{\mathbf{y}_{1:t-1},\mathbf{l}_{1:t-1}}(\mathbf{l}_t) = \mathbb{E}_{p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\mathbf{l}_{1:t})}\left[\mathcal{H}\left[p(\theta|\mathbf{y}_{1:t},\mathbf{l}_{1:t})\right]\right]. \tag{7}$$

which can be minimized at each time t.

A major downside of Bayesian experimental design is that calculating the expected posterior entropy, and therefore determining the optimal design, is analytically intractable for all but the simplest problems, and even approximations can be expensive. However, recent approaches to approximate the optimal solution by combining experimental design with tools from variational inference and deep learning [10], along with improvements in generative modelling [18, 19], have expanded the class of problems for which it is viable. These, and further innovations we introduce including the connection to structured dropout and probabilistic patch-based image retrieval, mean that experimental design can now be considered even when dealing with high-dimensional domains such as natural images.

Although the Shannon entropy in the posterior may seem like an arbitrary choice of utility function, it can be shown to be the optimal strategy to minimise a negative log-likelihood loss when the classifier is optimal for given attention locations. A proof of this is in the appendix. Note, however, that sequentially performing Bayesian optimal experimental design is a greedy approximation to the optimal strategy of minimising the expected entropy after multiple time steps.

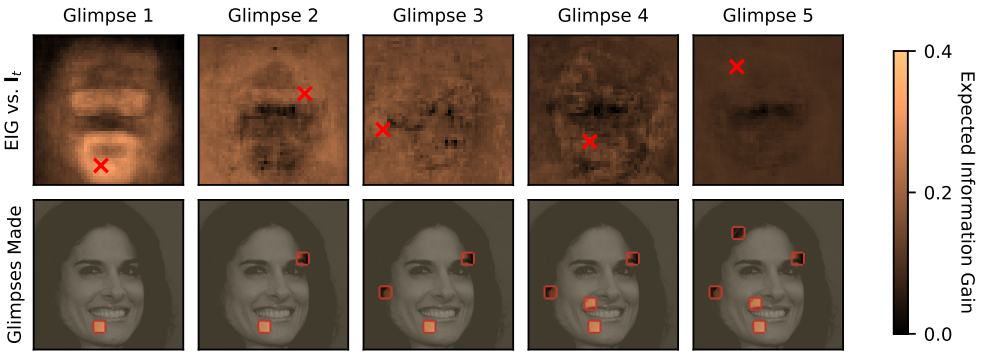

Figure 1: **Top row:** The estimated expected information gain as a function of the next glimpse location, during a sequence aimed at predicting the attribute 'Male'. The red cross marks the maximum of the expected information gain, which is subsequently attended to. **Bottom row:** A visualisation of the observed parts of the image after each glimpse. This is the information with which the expected posterior entropy at the next time is estimated. A representative sample of similar examples can be found in the appendix.

3 Method

At a high-level, our approach involves automatically annotating a subset of the training data with approximately optimal sequences of attention locations, derived by applying the methods described in the previous section. Since the optimal sequence of locations is dependent on the task, these sequences are computed separately for each attribute we wish to infer. They are then used to partially supervise the attention mechanism during training.

Experimental Design Procedure As discussed in Section 2, experimental design involves selecting a location to attend to, $\mathbf{l}_t$, at each time t by minimizing the expected posterior entropy given the previous locations and observations, $\mathbf{l}_{1:t-1}$ and $\mathbf{y}_{1:t-1}$ [22]:

$$\text{EPE}_{\mathbf{y}_{1:t-1},\mathbf{l}_{1:t-1}}(\mathbf{l}_t) = \mathbb{E}_{p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\mathbf{l}_{1:t})}\Big[\mathcal{H}\big[p(\theta|\mathbf{y}_{1:t},\mathbf{l}_{1:t})\big]\Big]. \tag{8}$$

Calculating this exactly is intractable, not least because we do not have an explicit model of θ and $\mathbf{y}$. We instead approximate it using a two-stage process: first, we draw approximate samples from $p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\mathbf{l}_{1:t})$. This is effectively conditional image sampling, for which we use a novel probabilistic image retrieval scheme. Then, for each sample, we compute $\mathcal{H}\left[q(\theta|\{\mathbf{y}_{1:t-1},\mathbf{y}_t^{(n)}\},\mathbf{l}_{1:t})\right]$, an amortized approximation of the posterior entropy. This yields a Monte Carlo estimate of Equation 8 of the form

$$\text{EPE}_{\mathbf{y}_{1:t-1},\mathbf{l}_{1:t-1}}(\mathbf{l}_t) \approx \frac{1}{N}\sum_{n=1}^{N}\mathcal{H}\big[q(\theta|\{\mathbf{y}_{1:t-1},\mathbf{y}_t^{(n)}\},\mathbf{l}_{1:t})\big], \qquad \mathbf{y}_t^{(n)} \sim r(\mathbf{y}_t|\mathbf{y}_{1:t-1},\mathbf{l}_{1:t}), \tag{9}$$

where $r(\mathbf{y}_t|\mathbf{y}_{1:t-1},\mathbf{l}_{1:t}) \approx p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\mathbf{l}_{1:t})$ is the distribution over images returned by the image retrieval scheme. We find $\mathbf{l}_t$ to minimise Equation 9 using a grid search over image locations. We now discuss the form of each of q and r.

Approximate Posterior Our approach involves estimating the posterior entropy for many possible values of $\mathbf{y}_t$. We do this by training a convolutional neural network (CNN) to map from a sequence of observations, $\mathbf{y}_{1:t}$, and locations, $\mathbf{l}_{1:t}$, to $q(\theta|\mathbf{y}_{1:t},\mathbf{l}_{1:t})$, an approximation of the generally intractable posterior. We then approximate the posterior entropy as the entropy of $q(\theta|\mathbf{y}_{1:t},\mathbf{l}_{1:t})$:

$$\mathcal{H}\left[p(\theta|\mathbf{y}_{1:t},\mathbf{l}_{1:t})\right] \approx \mathbb{E}_{q(\theta|\mathbf{y}_{1:t},\mathbf{l}_{1:t})}\left[-\log q(\theta|\mathbf{y}_{1:t},\mathbf{l}_{1:t})\right] \tag{10}$$

This approximation is inspired by the work of Foster et al. [10]. They make use of a similar approximation which provides an upper bound on the expected posterior entropy: $\mathbb{E}_{p(\theta,|\mathbf{y}_{1:t},\mathbf{l}_{1:t})}\left[-\log q(\theta|\mathbf{y}_{1:t},\mathbf{l}_{1:t})\right]$. Since we are unable to sample from $p(\theta|\mathbf{y}_{1:t-1},\mathbf{l}_{1:t-1})$, we

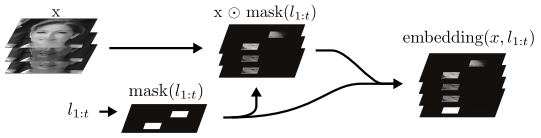

Figure 2: The embedding of $\mathbf{y}_{1:t}$ and $\mathbf{l}_{1:t}$. $\mathbf{l}_{1:t}$ is used to create a mask which conceals unobserved pixels. This mask is applied to the image and then concatenated as an additional channel.

can only use the approximation in Equation 10 which does not provide a bound. However, in both cases, $q(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})$ can be learned to make the approximation as close as possible, and it becomes exact when $q(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t}) = p(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})$. Key to the approach of Foster et al. [10] is learning a mapping from $\mathbf{y}_t$ to $q(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})$, allowing the sharing of information between "nearby" samples of $\mathbf{y}_t$ to massively reduce the computational cost of the experimental design. We take this further by amortizing over not just $\mathbf{y}_t$, but also the location under consideration, $\mathbf{l}_t$; previous glimpses and locations, $\mathbf{l}_{1:t-1}$ and $\mathbf{y}_{1:t-1}$; and even the number of previous glimpses, t.

We do this by proposing $q(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})$ from a CNN which receives an embedding of both $\mathbf{y}_{1:t}$ and $\mathbf{l}_{1:t}$. As in Foster et al. [10], the network is trained to minimise the expectation over $\mathbf{y}_t$ of the KL divergence between the posterior and the variational approximation. Additionally, since we want the approximation to be good for all t, $\mathbf{l}_{1:t}$ and $\mathbf{y}_{1:t-1}$, the loss used is an expectation over distributions of all of these: $p(t)$, $p(\mathbf{l}_{1:t}|t)$, and $p(\mathbf{y}_{1:t}|\mathbf{l}_{1:t})$. To weight all locations and times equally in the loss, uniform distributions are used: $p(t)$ is a uniform distribution over times $1, \ldots, T$ and $p(\mathbf{l}_{1:t}|t) = \prod_{i=1}^{t} p(\mathbf{l}_i)$ samples each $\mathbf{l}_i$ independently from a uniform distribution over all allowed locations. The gradient of this loss is estimated as follows:

$$\mathcal{L}_\phi = \mathbb{E}_{p(\mathbf{y}_{1:t}|\mathbf{l}_{1:t})p(\mathbf{l}_{1:t}|t)p(t)} \left[\mathrm{KL}\left(p(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})||q_\phi(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})\right)\right] \tag{11}$$

$$= \mathbb{E}_{p(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})p(\mathbf{y}_{1:t}|\mathbf{l}_{1:t})p(\mathbf{l}_{1:t}|t)p(t)} \left[\log \frac{p(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})}{q_\phi(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})}\right] \tag{12}$$

$$\frac{\partial \mathcal{L}_\phi}{\partial \phi} = \mathbb{E}_{p(\theta, \mathbf{y}_{1:t}|\mathbf{l}_{1:t})p(\mathbf{l}_{1:t}|t)p(t)} \left[-\frac{\partial}{\partial \phi} \log q_\phi(\theta|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})\right]. \tag{13}$$

Samples from this expectation can be taken by sampling from each of $p(t)$ and $p(\mathbf{l}_{1:t}|t)$, sampling an image and θ from the dataset, and deterministically extracting $\mathbf{y}_{1:t}$ from this image. The term inside the expectation can then be calculated using standard backpropagation.

In order to encode the relevant spatial information, and allow a single network to be used for varying t, we embed $\mathbf{y}_{1:t}$ and $\mathbf{l}_{1:t}$ as shown in Figure 2. The locations attended to, $\mathbf{l}_{1:t}$, are used to create a mask which is one at observed pixels and zero elsewhere. This mask is used to perform a structured "dropout" [11, 25] on the full image by setting all unobserved pixels to zero, and then concatenated to the image as an additional channel. The resulting tensor forms the input to the neural network. This embedding maintains the spatial information whilst enforcing an invariance to the order of the location sequence. An example of the expected information gains computed using this technique, in conjunction with the rest of our algorithm, is shown in Figure 1.

Conditional Image Sampling To estimate the expected posterior entropy for a given next glimpse location, we need some distribution over what will be seen there. Following Equation 9, this distribution is $r(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{l}_{1:t}) \approx p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{l}_{1:t})$. Under the model described in Section 2, where $\mathbf{y}_t$ is a deterministic function of $\mathbf{x}$ and $\mathbf{l}_t$ (i.e. $p(\mathbf{y}_t|\mathbf{x}, \mathbf{l}_t)$ is a Dirac delta distribution), we can sample from this by sampling an entire image, $\mathbf{x}$, conditioned on $\mathbf{y}_{1:t-1}$ and $\mathbf{l}_{1:t-1}$, and then simply extracting the patch at location $\mathbf{l}_t$. Since Equation 9 must be evaluated for multiple $\mathbf{l}_t$ to minimise it, this approach allows sharing of computation by sampling the same images, and then simply extracting different patches to condition on different $\mathbf{l}_t$. Figure 3 illustrates the distributions over the full image inferred at each time whilst generating an optimal sequence.

To approximate the posterior distribution over images, $p(\mathbf{x}|\mathbf{y}_{1:t-1}, \mathbf{l}_{1:t-1})$, we first model a joint distribution of images and observations, $\hat{p}(\mathbf{x}, \mathbf{y}_{1:t-1}|\mathbf{l}_{1:t-1}) = p(\mathbf{x}) \prod_{i=1}^{t-1} \hat{p}(\mathbf{y}_i|\mathbf{x}, \mathbf{l}_i)$. For the prior over images, $p(\mathbf{x})$, we use a generative adversarial network. Specifically, we use StyleGAN [18] trained on the CelebA-HQ [17] dataset. The observation model, $\hat{p}(\mathbf{y}_i|\mathbf{x}, \mathbf{l}_i)$, assumes $\mathbf{y}_i$ is distributed according to a Gaussian centred on the pixel values of the image at $\mathbf{l}_i$, with independent noise in

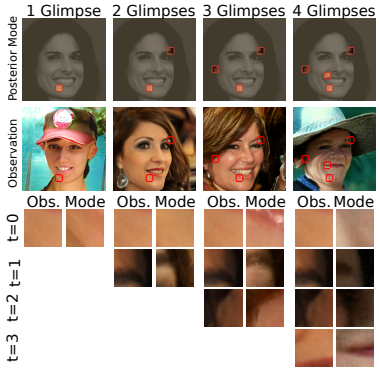

Figure 3: **Top row**: True image with the observed patches marked for the first four time steps of an experimental design procedure. **Second row:** Mode of the posterior distribution over the image inferred after each glimpse in the top row. **Rest:** Comparisons of the observations and the corresponding glimpses of the posterior mode at each time step.

each dimension. This joint distribution defines a posterior, $\hat{p}(\mathbf{x}|\mathbf{y}_{1:t-1}, \mathbf{l}_{1:t-1})$, which can be sampled from using techniques from Bayesian inference. Note that the addition of pixel-wise noise in the observation model differs from the model described in Section 2, where $p(\mathbf{y}_i|\mathbf{x}, \mathbf{l}_i)$ is a Dirac delta distribution. This relaxation was necessary to make inference feasible.

We initially attempted to use Hamiltonian Monte Carlo [8, 13] to sample from this distribution, but it was found to explore the distribution too slowly to be practical for our procedure. Instead, we use a scheme based on importance sampling [2]. This was designed with two major obstacles in mind: the prohibitive computational cost of repeatedly running the generative model and the equally prohibitive cost of loading a large number of images into memory. To avoid running the generative model during sampling, we first create a dataset of 1.5 million images from it. To avoid the need to load these images into memory during sampling, we use a novel technique (described in the paragraph below) based on image retrieval [15] to cheaply approximate the likelihood, given any observed glimpse sequence, of each image in the dataset. This is used to construct a proposal distribution over the images. Samples from this proposal are loaded into memory and reweighted according to the exact likelihood, $\prod_{i=1}^{t-1} \hat{p}(\mathbf{y}_i|\mathbf{x}, \mathbf{l}_i)$. These weighted samples then approximate the posterior over images [2], $\hat{p}(\mathbf{x}|\mathbf{y}_{1:t-1}, \mathbf{l}_{1:t-1})$. Since the number of times the neural network in Section 3 must be run is proportional to the number of images sampled, we draw a new, smaller, set of samples from this weighted approximation of the posterior [7]. Intuitively, this resampling step decreases the computation wasted on samples with low weights. A more detailed description of the probabilistic image retrieval algorithm, along with pseudocode, is available in the appendix.

We now outline our procedure for approximating, for each image in the dataset, the likelihood of an observed glimpse sequence. We pay the up-front cost of performing principal component analysis [16] on the dataset. This leads to an memory-efficient approximate representation of the dataset with a set of principal component vectors and an affine transformation which approximately reconstructs an image from its principal components. The affine property of this transformation makes it possible to cheaply construct an approximation of a small portion of the image (such as a glimpse location) without needing to construct the rest of the image. We then perform the likelihood calculation with these reconstructions as stand-ins for the patches, giving an approximate likelihood. A more detailed description is available in the appendix.

Training with supervision The experimental design procedure described above yields task-specific, near-optimal attention sequences. For a given task, we annotate a subset of the training data with these, and use them to partially supervise a neural attention mechanism while it is trained on the same task. This is done by augmenting the end-to-end loss with a term proportional to the negative log-likelihood of each location in the sequence under the attention mechanism's proposed distribution. This term is added for every training example for which annotations are available. Additionally, when the neural network is running on such a training example, the attention locations are fixed to those in the provided sequence. In addition to this loss on the annotated data, the REINFORCE algorithm [30]

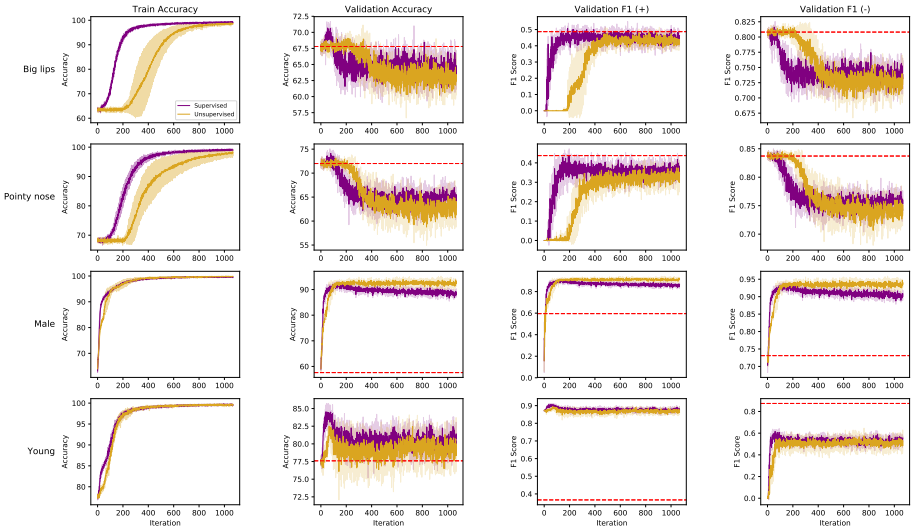

Figure 4: Training and validation loss throughout training with partially supervised attention (purple) and fully unsupervised attention (orange) for 4 binary attribute classification tasks. From top to bottom, we report results for "Big Lips", "Pointy Nose", "Male" and "Young". The first two columns report training and validation accuracy. The third column reports the F1 score for each class on the validation set and the fourth column reports the F1 score on the complement of each class. For the final three columns, the validation set is fixed and a baseline is shown in red for each metric as the best score obtainable by making the same prediction for every example.

is used on the rest of the training examples. A learned control variate is used to reduce the variance of the gradient estimate on these [21].

4 Experiments

To test whether our method of partial supervision improves training, we carry out experiments on 4 different facial attribute classification tasks using the the CelebA-HQ dataset [17]. This dataset contains 30 000 annotated images, which we split into a training set of 28 500 examples, a validation set of 500 examples and a test set of 1000 examples. We use an architecture with the structure described in Section 2 and train it using both partially supervised data (as described in Section 3), and fully unsupervised data (i.e. using REINFORCE [30]). For the partially supervised training, we create between 450 and 900 supervised examples for each task, comprising between 1.5% and 3.2% of the training data. The attention mechanism used takes glimpses of size 16×16 from the images rescaled to 224×224 resolution. Our approach leads to faster training on all tasks and higher test accuracy on three of the four tasks.

In Figure 4, we show training and validation losses for 4 attribute classification tasks over the course of 240 epochs, or 106 800 iterations of minibatch gradient descent. It can be seen that the semi-supervision leads to considerably faster training. Table 1 quantifies this speedup in terms of the number of iterations required to achieve 90% training accuracy (right columns). We observe quicker training in all cases, with the networks reaching this point $2.19\times$ faster on average, and generally with lower variance.

Additionally, when a supervision signal is provided through near-optimal sequences, we achieve higher accuracy on most tasks during both training and testing. In Table 1, we achieve higher accuracy on a held out test set for all attributes except Male by testing using the weights saved at the iteration with the greatest validation accuracy. Figure 5 helps explain the improvement by illustrating the fundamental differences between the distribution learned by the partially supervised and unsupervised

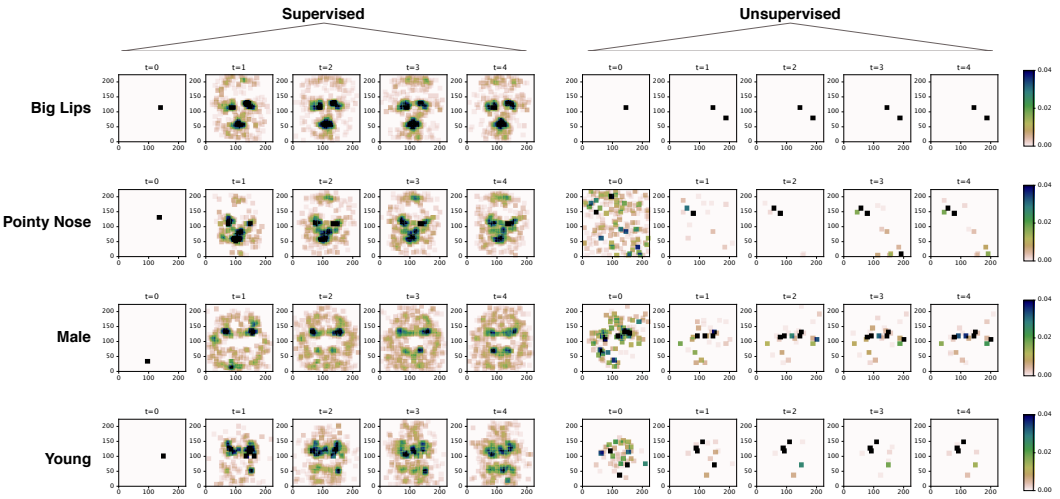

Figure 5: We show a density estimate of the locations attended to at each timestep by a fully trained network for each given task with (left) and without (right) supervision. This shows that the policies learned are indeed task-specific. The supervision leads to a much broader distribution over attention locations on all but the first timestep, where they learn the optimal policy of always attending to the same location. This is in contrast to the unsupervised attention, in which the support appears to narrow after the first timestep.

Table 1: Comparison of accuracy and training speed ($\pm$std)

	Test Accuracy (%)		Iterations ($/10^3$) to 90%	
Attribute	Supervised	Unsupervised	Supervised	Unsupervised
Big lips	$\mathbf{71.0 \pm 0.4}$	68.9 ± 0.1	$\mathbf{17.4 \pm 0.3}$	47.8 ± 9.6
Pointy nose	$\mathbf{70.3 \pm 0.1}$	69.8 ± 0.0	$\mathbf{25.3 \pm 1.9}$	45.3 ± 11.7
Young	$\mathbf{84.8 \pm 1.0}$	83.7 ± 0.5	$\mathbf{9.6 \pm 0.51}$	11.7 ± 0.51
Male	91.9 ± 0.1	$\mathbf{93.3 \pm 0.7}$	$\mathbf{3.3 \pm 0.17}$	6.9 ± 1.7

mechanisms. In fact, we found that the unsupervised mechanism would, in many cases, attend to the same location for each of the second to fifth glimpses (on 95.7% of test images for Big Lips, 0.9% for Male, 1.9% for Pointy Nose and 52.4% for Young). This contrasts to a maximum across tasks of 0.3% with semi-supervision. Since the network can obtain no new information by attending to the same position repeatedly, this cannot be an optimal policy from an information-theoretic perspective.

5 Discussion and Conclusions

We have shown that using our novel Bayesian optimal experimental design techniques to generate approximately optimal attention sequences can greatly improve the training of hard attention neural networks through semi-supervision. We showed that our approach can be used to derive bespoke glimpse policies for different classification tasks. Our framework is capable of being extended in future work to tasks such as question answering where the latent variable of interest is more richly structured. It can also be more immediately improved by making predictions when the classification entropy is sufficiently low, rather than always making T glimpses. The expected information gains shown in Figure 1 suggest that in this example, even with the small glimpse regions we use, there is little new information to gain after the first couple of glimpses. More glimpses in such situations simply waste computation.

Our methodology requires either a very fast generative model that can be used for image completion from glimpsed regions or a dataset large enough to maintain a high sample diversity when images are constrained by increasingly many glimpsed regions. Nonparametric density representations, as we

used, are unlikely to be effective when the model underlying the experimental design optimization becomes more complex. In this way our work is an ideal consumer of continuing research into efficient image completion techniques.

While we have not stressed this throughout, the particular model of attention we employ is intentionally compatible with the way real-world camera sensor systems work since we access only a single subregion of the image at a time. Foveated approaches to attention require access to the entire image at varying resolutions which, on current production cameras, can only be accomplished by averaging over the whole image repeatedly, losing whatever computational advantage they might otherwise accrue.

References

[1] John R Anderson. *Cognitive psychology and its implications*. Macmillan, 2005.

[2] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[3] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.

[4] Mark F Bear, Barry W Connors, and Michael A Paradiso. *Neuroscience*, volume 2. Lippincott Williams & Wilkins, 2007.

[5] Neil DB Bruce and John K Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of vision*, 9(3):5–5, 2009.

[6] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

[7] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE, 2005.

[8] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[9] Harriet Feldman and Karl Friston. Attention, uncertainty, and free-energy. *Frontiers in human neuroscience*, 4:215, 2010.

[10] Adam Foster, Martin Jankowiak, Eli Bingham, Paul Horsfall, Yee Whye Teh, Tom Rainforth, and Noah Goodman. Variational estimators for bayesian optimal experimental design. *arXiv preprint arXiv:1903.05480*, 2019.

[11] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

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

[13] Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[14] Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. *Vision research*, 49 (10):1295–1306, 2009.

[15] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Improving bag-of-features for large scale image search. *International journal of computer vision*, 87(3):316–336, 2010.

[16] Ian Jolliffe. *Principal component analysis*. Springer, 2011.

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

[19] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

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

[21] John Paisley, David Blei, and Michael Jordan. Variational bayesian inference with stochastic search. *arXiv preprint arXiv:1206.6430*, 2012.

[22] Tom Rainforth. *Automating Inference, Learning, and Design using Probabilistic Programming*. PhD thesis, 2017.

[23] Ronald A Rensink. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42, 2000.

[24] Philipp Schwartenbeck, Thomas FitzGerald, Ray Dolan, and Karl Friston. Exploration, novelty, surprise, and free energy minimization. *Frontiers in psychology*, 4:710, 2013.

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[26] Harry Moss Traquair. *An introduction to clinical perimetry*. Mosby, 1949.

[27] George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2627–2636, 2017.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[29] Donald P Warwick and Charles A Lininger. *The sample survey: Theory and practice.* McGraw-Hill, 1975.

[30] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[31] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

Appendix A: Proof - Minimising EPE Minimises NLL

For a perfect classifier which outputs a distribution $q(\theta|\mathbf{y}, \mathbf{l})$, this distribution is exactly equal to the true posterior distribution, $p(\theta|\mathbf{y}, \mathbf{l})$. Therefore, the expectation of the negative log-likelihood over a data distribution, $p(\theta, \mathbf{y}|\mathbf{l})$, is given by:

$$\mathbb{E}_{p(\theta,\mathbf{y}|\mathbf{l})}\big[-\log q(\theta|\mathbf{y}, \mathbf{l})\big] \tag{1}$$

$$=\mathbb{E}_{p(\theta,\mathbf{y}|\mathbf{l})}\big[-\log p(\theta|\mathbf{y}, \mathbf{l})\big] \tag{2}$$

$$=\mathbb{E}_{p(\mathbf{y}|\mathbf{l})}\big[\mathcal{H}\left[p(\theta|\mathbf{y}, \mathbf{l})\right]\big] \tag{3}$$

which is the expected posterior entropy. This implies that selecting $\mathbf{l}$ to minimise this will minimise a negative log-likelihood loss if $q(\theta|\mathbf{y}, \mathbf{l})$ is able to fit perfectly to $p(\theta|\mathbf{y}, \mathbf{l})$.

Appendix B: Conditional Image Sampling Details

Algorithm 1 SAMPLEIMAGES outlines our image sampling procedure. In SAMPLEPROPOSAL, K_1 images are sampled from a cheap approximation of $p(\mathbf{x}|\mathbf{y}_{1:t}, \mathbf{l}_{1:t})$ using weights calculated with a PCA-approximation of the observed patches. The sampled images are loaded into memory and exact weights are computed. K_2 images are then sampled and returned from the resulting categorical distribution. RESAMPLE normalises the weights and uses them to sample new indices using residual resampling [4].

procedure SAMPLEPROPOSAL($\mathbf{x}, \mathbf{l}_{1:t}$)
  Select relevant weight matrix columns $\tilde{\mathbf{W}} \leftarrow \text{SLICE}(\mathbf{W}, \mathbf{l}_{1:t})$
  Select relevant weight matrix columns $\tilde{\mu} \leftarrow \text{SLICE}(\mu, \mathbf{l}_{1:t})$
  Reconstruct observations $\hat{\mathbf{y}}_{1:t} \leftarrow \tilde{\mu} + \tilde{\mathbf{W}}^\top \mathbf{W}\mathbf{x}$
  for $i = 1, \ldots, N$ **do**
    Approximate observed patches $\hat{\mathbf{y}}^i_{1:t} \leftarrow \tilde{\mathbf{W}}^\top \mathbf{z}^i$
    Compute approximate likelihood $w_1^i \leftarrow \mathcal{N}\left(\hat{\mathbf{y}}_{1:t}|\hat{\mathbf{y}}^i_{1:t}, \sigma^2\right)$
  $j^{(1)}, \ldots, j^{(K_1)} \leftarrow \text{RESAMPLE}(w_1^1, \ldots, w_1^N)$
  return $\{i^{(k)}, w_1^{i^{(k)}}\}$ for $k = 1, \ldots, K_1$

procedure SAMPLEIMAGES($\mathbf{x}, \mathbf{l}_{1:t}$)
  $\{i^{(1)}, w_1^{i^{(1)}}\}, \ldots, \{i^{(K_1)}, w_1^{i^{(K_1)}}\} = \text{SAMPLEPROPOSAL}(\mathbf{x}, \mathbf{l}_{1:t})$
  for $k = 1, \ldots, K_1$ **do**
    Load $\mathbf{x}^{i^{(k)}}$
    $\mathbf{y}^k_{1:t} \leftarrow \text{Glimpse}(\mathbf{x}^{i^{(k)}}, \mathbf{l}_{1:t})$
    Compute exact likelihood $p(\mathbf{y}_{1:t}|\mathbf{x}^{i^{(k)}}, \mathbf{l}_{1:t}) = \mathcal{N}\left(\mathbf{y}_{1:t}|\mathbf{y}^k_{1:t}, \sigma^2\right)$
    Compute weight $w_2^{(k)} \leftarrow \frac{p(\mathbf{y}_{1:t}|\mathbf{x}^{i^{(k)}}, \mathbf{l}_{1:t})}{w_1^{i^{(k)}}}$
  $j^{(1)}, \ldots, j^{(K_2)} \leftarrow \text{RESAMPLE}(w_2^1, \ldots, w_2^{K_1})$
  return $\mathbf{x}^{j^{(k)}}$ for $k = 1, \ldots, K_2$

The first step of this procedure defines a proposal distribution using the dataset, which is distributed according to our generative model, $p(\mathbf{x})$. For each image in the dataset, we can generate triples of the image, the observed portion, and the PCA reconstruction of the observed portion given $\mathbf{l}_{1:t-1}$: $(\mathbf{x}, \mathbf{y}_{1:t-1}, \hat{\mathbf{y}}_{1:t-1})$. Weighting these (as well as their horizontally flipped counterparts as a form of datset augmentation) according to an approximation of their likelihood, $w_1^i = \mathcal{N}(\hat{\mathbf{y}}_{1:t}|\hat{\mathbf{y}}^i_{1:t}, \sigma^2)$, and normalising the weights yields a categorical distribution approximating $\hat{p}(\mathbf{x}|\hat{\mathbf{y}}_{1:t})$ [1]. This then becomes a proposal distribution with which we perform importance sampling on $\hat{p}(\mathbf{x}|\hat{\mathbf{y}}_{1:t})$. This is done by drawing samples $\mathbf{x}^{(k)}$ for $k = 1, \ldots, K_1$ with residual resampling [4]. Each sample is then weighted with $w_2^{(k)} = \frac{\mathcal{N}(\mathbf{y}_{1:t}|\mathbf{y}^{i^{(k)}}_{1:t}, \sigma^2)}{w_1^{i^{(k)}}}$. These weights give an approximation of the posterior of the image under the Gaussian noise assumption, which becomes increasingly close as the dataset size and K_1 tend to ∞. Finally, to reduce the number of samples for which the posterior entropy must be approximated, we resample K_2 samples from the resulting categorical distribution with residual resampling.

Principal component analysis allows for a memory-efficient representation of the dataset through a mean image, μ, a low-dimensional vector for each image, $\mathbf{z}^i$, and an orthogonal matrix, $\mathbf{W}$, which transforms from an image, $\mathbf{x}^i$, into the corresponding $\mathbf{z}^i$ as follows:

$$\mathbf{z}^i = \mathbf{W}\mathbf{x}^i \tag{4}$$

Denoting the dimensionality of $\mathbf{z}$ as L, the number of images as N, and the number of pixels per image as P, these objects can be stored with memory complexity $\Theta(NL+PL)$, compared to $\Theta(NP)$ for the entire dataset. Since $\mathbf{W}$ is orthogonal, image i can be approximated using $\mathbf{z}^i$ as

$$\hat{\mathbf{x}}^i = \mathbf{W}^\top \mathbf{z}^i \tag{5}$$

and $\hat{\mathbf{x}} \approx \mathbf{x}$ for large enough L. If only certain pixels of the reconstructed image are required, $\mathbf{x}^i_{p_1:p_C}$ these can be obtained efficiently by using $\tilde{\mathbf{W}}^\top$, a matrix made up of rows p_1 to p_C of $\mathbf{W}^\top$:

$$\mathbf{x}^i_{p_1:p_C} = \tilde{\mathbf{W}}^\top \mathbf{z}^i. \tag{6}$$

This allows us to construct an approximation of the observed portion with each image in the dataset in a time and memory-efficient manner. Additionally, we found that our proposal was improved when the approximate likelihood was calculated with a reconstruction of the observations as $\hat{\mathbf{y}}_{1:t} = \tilde{\mathbf{W}}^\top \mathbf{W}\mathbf{x}$, rather than the true observations $\mathbf{y}_{1:t}$. Although this requires access to the full true image, this is acceptable as the full image was always available when we carried out our experimental design. Also, since this is only used to calculate the proposal distribution, it should have limited effect on the samples returned after new weights are calculated with the exact likelihood.

Appendix C: Architecture and Hyperparameters

Entropy Approximation

For the CNN which gives the varitaional approximation to the entropy, we use the DenseNet-121 architecture [5] pre-trained on ImageNet [3]. It is modified to accept an additional input channel, for which the weights are set to zero. The last layer is also replaced with a linear layer which parameterises Bernoulli distributions over all 40 labelled attributes in the CelebA-HQ dataset.

Conditional Image Sampling

We use parameters $K_1 = 1000$ and $K_2 = 200$ for the image retrieval, and a 256-dimensional principal component vector. The observation noise distribution has standard deviation increasing with the number of glimpses conditioned on: it is 5 after 1 glimpse; 10 after 2 glimpses; 20 after 3 glimpses; and 40 after 4 glimpses. These variances are for images with pixel values normalised to have zero mean and unit standard deviation. In our experiments, the use of these variances led to effective sample sizes that were, on average, approximately 30 when conditioning on 1 glimpse and approximately 10 when conditioning on 2, 3, or 4 glimpses (as measured by the inverse of the sum of the squared normalized weights [6]).

Experimental Hyperparameters

The glimpses are 16×16 squares of pixels, taken from the images resized to 224×224 resolution. The neural network proposes a categorical distribution over a 50×50 grid of locations at each time step. We use a GRU [2] with a 256-dimensional state as the core of the recurrent neural network. The glimpses are embedded into a 128-dimensional vector by 6 convolutional layers with up to 64 channels, kernels of size 3 and ReLU activations, along with 2 max-pooling layers. Two fully connected layers then transform this into a 256-dimensional vector. This, added to a 256-dimensional embedding of the location created by two fully connected layers, formed the input to the GRU. The GRU output was transformed into the distribution over locations through two fully connected layers, and into the inferred attribute distribution with a single fully connected layer. The batch size used during training is 64 for both the semi-supervised and unsupervised case.

We train the networks using: 850 near-optimal sequences for attribute "Big Lips"; 450 for "Pointy Nose"; 700 for "Male"; and 600 for "Young". These are calculated for examples in the training set alone.

Appendix D: Estimated Mutual Informations

Figures 1 and 2 show representative sequences of near-optimal trajectories, along with visualizations of the estimated expected posterior entropies used to calculate them. Note that these figures show the expected posterior entropy for each glimpse location, whereas Figure 1 of the paper showed estimates of the mutual information, which were made by subtracting the expected posterior entropy from an estimate of the current entropy. Correspondingly, glimpses are taken at the minima rather than the maxima. The expected posterior entropy, as shown here, is calculated and minimised whilst making training data, whereas the mutual information was shown simply for visualisation.

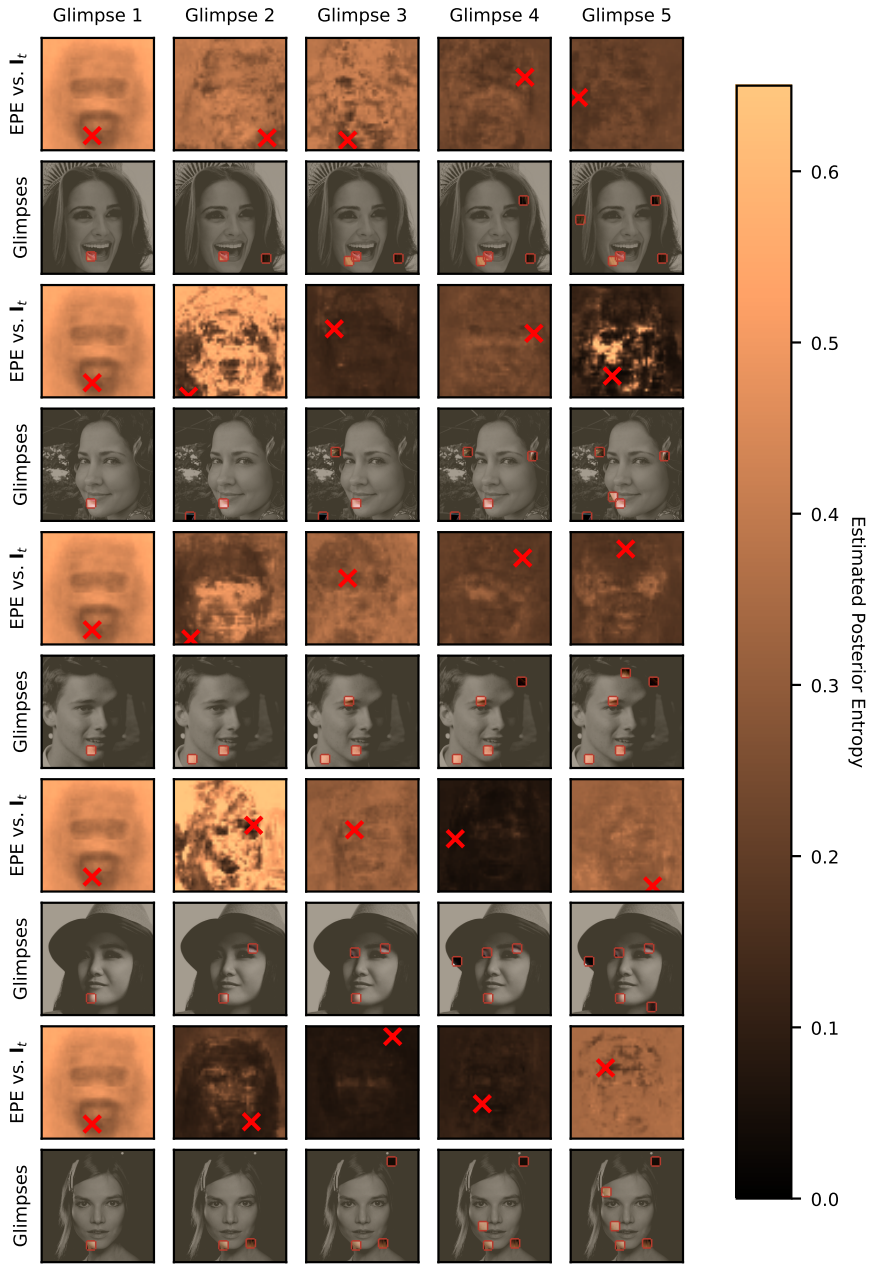

Figure 1: Expected posterior entropy given each location considered during the sequential experimental design procedure (for determining attribute 'Male') for a representative sample of images. Figure 2 contains further examples.

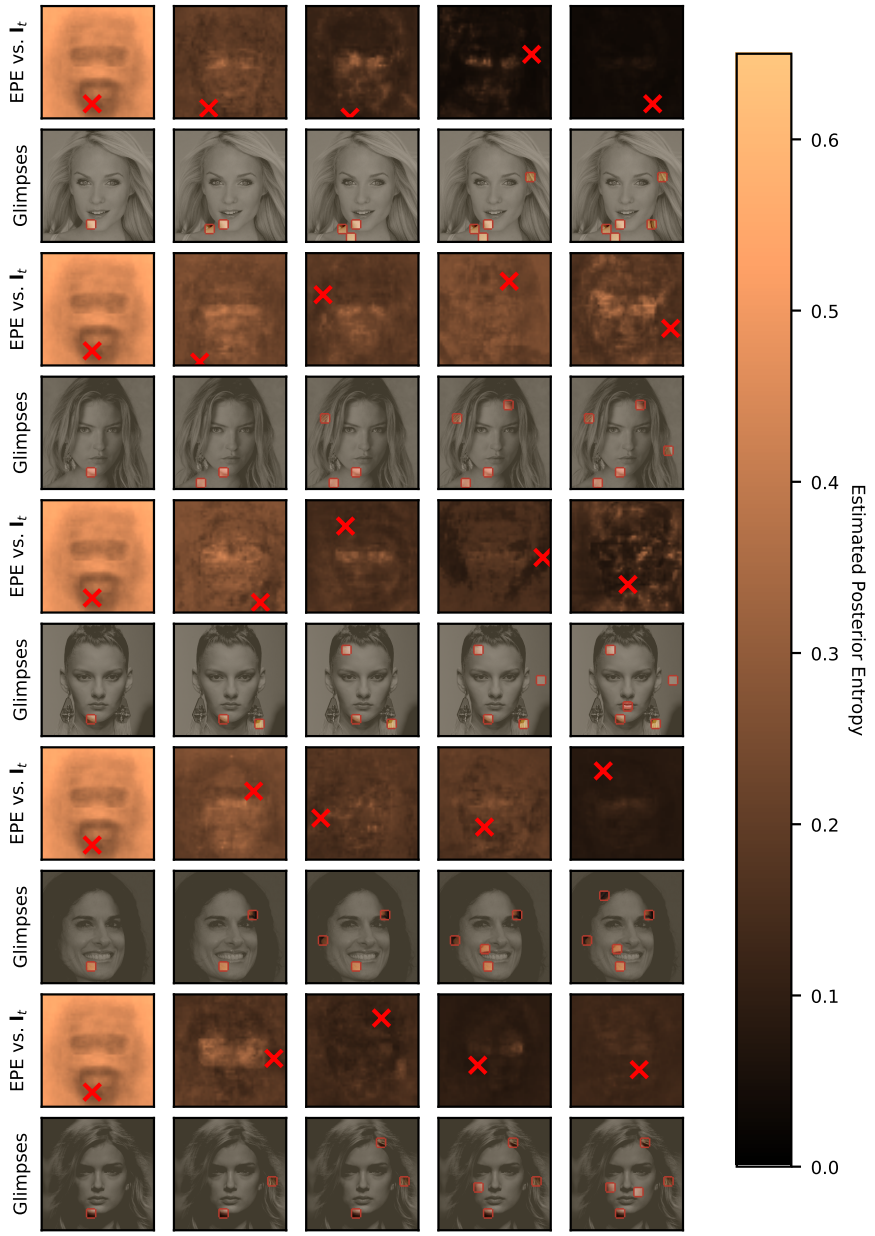

Figure 2: More samples similar to Figure 1. Expected posterior entropy at each location considered during the sequential experimental design procedure for determining attribute 'Male' on a representative sample of images.

Appendix E: Image Samples

Figures 3 and 4 show representative samples from the probabilistic image retrieval process during the creation of optimal sequences for class 'Male'. It can be seen that the image sampling is able to capture the rough structure of the image reasonably well by matching low-frequency patterns at the glimpse locations, but it is less well-suited to matching finer details. This may be a result of the modelling assumption of independent pixel-wise Gaussian noise.

Figure 3: Visualisations of the conditional image sampling. The top row of each shows the true image, and the observed patches at each time step. The second row shows the mode of the posterior conditioned on these glimpses. The rows below show close-ups of the 16×16 glimpses on which the samples are conditioned (for both the original image and the mode).

Figure 4: More visualisations of the conditional sampling procedure, following from Figure 3.

References

[1] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[4] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE, 2005.

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[6] Leslie Kish. Cumulating/combining population surveys. *Survey Methodology*, 25(2):129–138, 1999.