
Discrete flow posteriors for variational inference in discrete dynamical systems

Laurence Aitchison
University of Cambridge
Cambridge, CB2 1PZ, UK
laurence.aitchison@gmail.com

Vincent Adam
PROWLER.io
66-68 Hills Road, Cambridge, CB2 1LA, UK
vincent.adam@prowler.io

Srinivas C. Turaga
HHMI Janelia Research Campus
Ashburn, VA 20147
turagas@janelia.hhmi.org

Abstract

Each training step for a variational autoencoder (VAE) requires us to sample from the approximate posterior, so we usually choose simple (e.g. factorised) approximate posteriors in which sampling is an efficient computation that fully exploits GPU parallelism. However, such simple approximate posteriors are often insufficient, as they eliminate statistical dependencies in the posterior. While it is possible to use normalizing flow approximate posteriors for continuous latents, some problems have discrete latents and strong statistical dependencies, including the neuroscience problem of inferring discrete spiking activity from noisy calcium-imaging data: not only does the posterior inherit dependencies from synaptic connectivity encoded in the prior, but there are also strong explaining-away effects. The most natural approach to model these dependencies is an autoregressive distribution, but sampling from such distributions is inherently sequential and thus slow. We develop a fast, parallel sampling procedure for autoregressive distributions based on fixed-point iterations which enables efficient and accurate variational inference in discrete state-space latent variable dynamical systems. To optimize the variational bound, we considered two ways to evaluate probabilities: inserting the relaxed samples directly into the pmf for the discrete distribution, or converting to continuous logistic latent variables and interpreting the K-step fixed-point iterations as a normalizing flow. We found that converting to continuous latent variables gave considerable additional scope for mismatch between the true and approximate posteriors, which resulted in biased inferences, we thus used the former approach. Using our fast sampling procedure, we were able to realize the benefits of correlated posteriors, including accurate uncertainty estimates for one cell, and accurate connectivity estimates for multiple cells, in an order of magnitude less time.

1 Introduction

The development of variational auto-encoders (VAE) [1, 2] has enabled Bayesian methods to be applied to very large-scale data, by leveraging neural networks, trained by stochastic gradient ascent, to approximate the posterior. To train these neural network approximate posteriors, we need to sample from them in each stochastic gradient ascent step, restricting us to approximate posteriors in which sampling can be performed rapidly, by leveraging parallel GPU computations. As such,

most work on VAE’s has used factorised approximate posteriors (e.g. [1–5]), but in many domains of interest we expect the posterior over latents to be highly correlated, not only because the posterior inherits correlations from the prior (e.g. as we might find in a dynamical system), but also because the likelihood itself can induce correlations due to effects such as explaining away [6, 7]. One approach to introducing correlations into the approximate posterior is normalizing flows [8, 9] which transforms variables generated from a simple, often factorised distribution into a complex correlated distribution, in such a way that the determinant of the Jacobian (and thus the probability of the transformed variables) can easily be computed.

However, the normalizing flow approach can only be applied to continuous latents, and there are important problems which require discrete latent variables and correlated posteriors, making efficient and accurate stochastic variational inference challenging. In particular, we consider the neuroscience problem of inferring the correlated spiking activity of neural populations recorded by calcium imaging. Due to the indirect nature of calcium imaging, spike inference algorithms must be used to infer the underlying neural spiking activity leading to measured fluorescence dynamics [10, 11]. Not only does the data naturally induce strong explaining away induced anticorrelations (as a spike in nearby timebins produces very similar data), but there are prior correlations induced by synaptic connectivity which induces similar correlations in the approximate posterior.

To address these challenging tasks with discrete latent variables, and correlated approximate posteriors, we considered two approaches. First, we considered applying normalizing flows by transforming our discrete latents into continuous latents, which are thresholded to recover the original discrete variables [12]. However, we found that working with continuous variables gave rise to far more scope for mismatch between the approximate and true posteriors than working with discrete variables, and that this mismatch resulted in biased inferences. Instead, we developed a fast-sampling procedure for discrete autoregressive posteriors. Instead, we took the more natural approach: considering an autoregressive approximate posterior that mirrors the structure of the prior, but to circumvent the slow sequential sampling procedure, we developed flow-like fixed-point iterations that are guaranteed to sample the true posterior after T iterations, but in practice converge much more rapidly (in our simulations, ~ 5 iterations), and efficiently exploit GPU parallelism.

Applying the flow-like fixed point iterations to simulated neuroscience problems, we were able to sample from autoregressive approximate posteriors in almost the same time required for factorised posteriors, and at least an order of magnitude faster than sequentially sampling from the underlying autoregressive process, allowing us to realize the benefits of correlated posteriors in large-scale settings.

2 Background

The evidence lower bound objective (ELBO), takes the form of an expectation computed over the approximate posterior, $Q_\phi(z)$,

$$\mathcal{L} = E_{Q_\phi(z)} [\log P_\theta(x|z) + \log P_\theta(z) - \log Q_\phi(z)] \quad (1)$$

Optimizing this objective with respect to the generative model parameters, θ , is straightforward, as we can push the derivative inside the expectation, and perform stochastic gradient descent. However, we cannot do the same with the recognition parameters, ϕ , as they control the distribution over which the expectation is taken. To solve this issue, the usual approach is the reparameterisation trick [1, 2] which performs the expectation over IID random noise, and transforms this noise into samples from the approximate posterior,

$$\mathcal{L} = E_\epsilon [\log P_\theta(x, z(\epsilon; \phi)) - \log Q_\phi(z(\epsilon; \phi))]. \quad (2)$$

As the recognition parameters ϕ no longer appear in the distribution over which the expectation is taken, we can again optimize this expression using stochastic gradient descent.

While the reparameterisation trick is extremely effective for continuous latent variables, it cannot be used for discrete latents, as we cannot back-propagate gradients through discrete latents. To rectify this issue, one approach is to relax the discrete variables, \bar{z} , to form an approximately equivalent model with continuous random variables, \hat{z} , through which gradients can be propagated [12, 13]. To consider the simplest possible case, we take a single binary variable, \bar{z} , drawn from a Bernoulli distribution with log-odds ratio u , and probability $p = \sigma(u) = 1/(1 + e^{-u})$,

$$\bar{P}(\bar{z}) = \text{Bernoulli}(\sigma(u)), \quad (3)$$

Instead of sampling \bar{z} directly from a Bernoulli, we can obtain samples of \bar{z} from the same distribution by first sampling from a Logistic, and thresholding that sample using,

$$P(l) = \text{Logistic}(u, 1), \quad \bar{z} = \Theta(l) = \lim_{\beta \rightarrow \infty} \sigma(\beta l), \quad \hat{z} = \sigma(\beta l), \quad (4)$$

where $\Theta(l)$ is the Heaviside step function, which is 0 for negative inputs, and 1 for positive inputs, and β is an inverse-temperature parameter. For a non-infinite temperature, we obtain an analogous relaxed variable, \hat{z} , that lies between 0 and 1.

Notably, for any (directed) model with discrete latents, including priors and approximate posteriors in the VAE setting, we can always use this approach to find an equivalent model with thresholded continuous latent variables.

While we can usually use the same functional form for the likelihood for relaxed and discrete models, it is less clear how we should compute the probabilities of the relaxed discrete latent variables under the prior and approximate posterior. In particular, we have two options for how we evaluate probabilities for the relaxed variables.

2.1 Evaluating probabilities under the discrete model

The most straightforward approach is to simply insert the relaxed variables, \hat{z} into the original probability mass function for the discrete model, $\bar{P}(\hat{z})$, and $\bar{Q}(\hat{z})$ [13]. Taking the univariate example, this gives,

$$\log \bar{P}(\hat{z}) = \hat{z} \log p + (1 - \hat{z}) \log(1 - p). \quad (5)$$

However, this immediately highlights a key issue: to obtain a valid variational bound, we need to evaluate the probability density of samples from the approximate posterior. In our case, samples from the approximate posterior are the relaxed variables, \hat{z} , so we need to evaluate $\bar{Q}(\hat{z})$. However, we are actually using a different expression, $\bar{Q}(\hat{z})$, and while this may in practice be an effective approximation, it cannot give us a valid variational bound and therefore we must be (and are) careful to evaluate our model using $\bar{Q}(\bar{z})$, which we can compute but not differentiate.

2.2 Evaluating probabilities under the continuous model

To obtain a valid variational bound, we need to evaluate the actual probability of the relaxed variable, i.e. we need to compute $\hat{Q}(\hat{z})$, and $\hat{P}(\hat{z})$. While we could work with the \hat{z} directly, this is known to be numerically unstable so instead we work in terms of the logistic variables, l [12]. These two approaches are equivalent, in the sense that the ratio of prior and approximate posterior probabilities is the same, because the gradient terms introduced by the change of variables cancel. To understand how we set the distributions, $\hat{P}(l)$, and $\hat{Q}(l)$, we remember that the ultimate goal is to have a relaxed, continuous model which closely approximates the discrete model of interest. As such, we remember that we could obtain an exactly equivalent model by sending the inverse-temperature to ∞ , and using,

$$P(l) = \text{Logistic}(u; 1) \quad Q(l) = \text{Logistic}(v; 1). \quad (6)$$

Further, note that we have not specified whether these distributions correspond to the discrete or relaxed model, as we can obtain either \hat{z} or \bar{z} from exactly the same l , by using different inverse-temperatures for the transformation (Eq. 4).

3 Results

We are interested in discrete dynamical systems, which can be written directly in terms of the discrete variables, or equivalently in terms of continuous variables drawn from a Logistic distribution,

$$\bar{P}(\bar{\mathbf{z}}_t | \bar{\mathbf{z}}_{1:t-1}) = \text{Bernoulli}(\boldsymbol{\sigma}(\mathbf{u}_t)), \quad P(\mathbf{l}_t | \mathbf{l}_{1:t-1}) = \text{Logistic}(\mathbf{u}_t, 1) \quad (7a)$$

$$\mathbf{u}_t = \mathbf{u}_t(\bar{\mathbf{z}}_{1:t-1}) = \lim_{\beta \rightarrow \infty} \boldsymbol{\sigma}(\beta \mathbf{l}_{1:t-1}). \quad (7b)$$

To form the approximate posterior, the most straightforward approach is to use another discrete autoregressive process, as this allows us to effectively model any correlations induced by the prior,

$$\bar{Q}(\bar{\mathbf{z}}_t | \mathbf{x}, \bar{\mathbf{z}}_{1:t-1}) = \text{Bernoulli}(\boldsymbol{\sigma}(\mathbf{v}_t)), \quad Q(\mathbf{l}_t | \mathbf{x}, \mathbf{l}_{1:t-1}) = \text{Logistic}(\mathbf{v}_t, 1) \quad (8a)$$

$$\mathbf{v}_t = \mathbf{v}_t(\mathbf{x}, \bar{\mathbf{z}}_{1:t-1}) = \lim_{\beta \rightarrow \infty} \mathbf{v}(\mathbf{x}, \boldsymbol{\sigma}(\beta \mathbf{l}_{1:t-1})). \quad (8b)$$

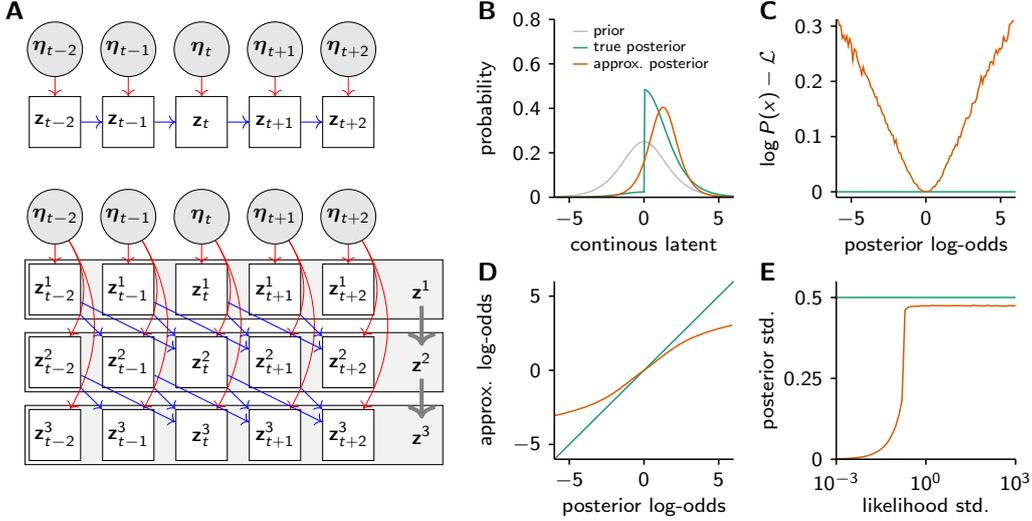


Figure 1: Differences between autoregressive and flow posteriors, based on discrete or continuous latents. **A** The sequential autoregressive (top), and parallel flow-like sampling procedure (bottom). **B** The true and approximate posteriors for continuous Logistic latent variables. **C** The difference between the model evidence and the ELBO induced by the posterior mismatch in **B**. **D** Biased inferences induced by the mismatch between the prior and approximate posterior. **E** The standard deviation of \hat{z} as we modulate the standard deviation of a Gaussian likelihood encouraging $\hat{z} \approx 0.5$.

3.1 A flow-like sampling procedure for discrete autoregressive distributions

While we can sample the autoregressive approximate posterior given above, in practice, this can be extremely slow for either the discrete or relaxed model, as the sequential structure (Fig. 1A top) fails to fully exploit the parallelism available on today’s GPU hardware. As such, we considered fixed-point iterations analogous to normalizing flows which do fully utilize GPU parallelism, and rapidly converge to a true sample (Fig. 1A bottom). In particular, these iterations begin by sampling IID noise, $\boldsymbol{\eta}$, from Logistic $(0, 1)$, and iterating,

$$\boldsymbol{l}^{k+1}(\boldsymbol{\eta}, \boldsymbol{l}^k) = \boldsymbol{\eta} + \mathbf{v}(\mathbf{x}, \boldsymbol{\sigma}(\beta \boldsymbol{l}^k)). \quad (9)$$

where all the variables are $N \times T$ matrices, and where $\mathbf{v}(\mathbf{x}, \boldsymbol{\sigma}(\beta \boldsymbol{l}^k))$ is simply Eq. (8b) computed in parallel across time for an externally specified input, rather than previous samples,

$$\boldsymbol{l}_t^{k+1}(\boldsymbol{\eta}_t, \boldsymbol{l}_{1:t-1}^k) = \boldsymbol{\eta}_t + \mathbf{v}_t(\mathbf{x}, \boldsymbol{\sigma}(\beta \boldsymbol{l}_{1:t-1}^k)). \quad (10)$$

We could use any initial condition, but the most straightforward is given by $z_{it}^0 = 0$, or equivalently, $l_{it}^0 = -\infty$. Importantly, these iterations are guaranteed to sample from the original autoregressive model if we run as many iterations as there are time-steps ($K = T$). In particular, we can see that the result of the first iteration, \boldsymbol{l}^1 , is correct at timestep 1, i.e. $\boldsymbol{l}_1 = \boldsymbol{l}_1^{k \geq 1}$, as \mathbf{v}_1 takes input only from data, \mathbf{x} . Further, the result of the second iteration is correct up to timestep 2, as \mathbf{v}_2 only takes input from \boldsymbol{l}_1^1 , and this is correct (i.e. it equals \boldsymbol{l}_1). In general, we have $\boldsymbol{l}_t = \boldsymbol{l}_t^{k \geq t}$. While this worst-case is extremely slow, converting a $\mathcal{O}(T)$ computation to an order $\mathcal{O}(T^2)$ computation, in practice, the iterations reach steady-state rapidly so we are able to use $K \ll T$, giving an order of magnitude improvement in efficiency by making improved use of GPU parallelism.

While this section gives a fast procedure for sampling from the approximate posterior, it is still unclear how we should evaluate probabilities under this approximate posterior. We have two options, which mirror the two options introduced above (Sec. 2.1 and Sec. 2.2 respectively).

3.2 Evaluating probabilities under the discrete model

First, we can compute relaxed Bernoulli variables using $\hat{\mathbf{z}} = \boldsymbol{\sigma}(\beta \boldsymbol{l}^K)$, and insert them into the probability mass function for the autoregressive discrete approximate posterior (Eq. 8a). While this can be computed efficiently in parallel, it introduces another level of mismatch between distribution

we sample and the distribution under which we evaluate the probability, in the sense that we sample using the fixed point iterations, but evaluate probabilities under the sequential autoregressive process (Eq. 8a). Remarkably, this is often not an issue for the discrete model, as we can simply iterate until convergence, and convergence is very well-defined as the latents are either 0 or 1. However, for the relaxed model it is more difficult to define convergence as the latents can lie anywhere between 0 and 1, so in practice we used a fixed number of iterations (in particular, $K = 5$).

3.3 Evaluating probabilities under the continuous model

To evaluate the probability of the continuous variables under the fixed point iterations, we interpret the iterations as constituting a normalizing flow.

Normalizing flows exploit the fact that we can compute the probability density of a random variable, $\mathbf{l}^k(\boldsymbol{\eta})$, generated by transforming, via a one-to-one function, a sample $\boldsymbol{\eta}$,

$$P(\mathbf{l}^k(\boldsymbol{\eta})) = P(\boldsymbol{\eta}) \left| \frac{\partial \mathbf{l}^k(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \right|^{-1} \quad (11)$$

where $|\partial \mathbf{l}^k / \partial \boldsymbol{\eta}|$ is the absolute value of the determinant of the Jacobian of $\mathbf{l}^k(\boldsymbol{\eta})$. While the determinant of the Jacobian is often very difficult to compute, we can ensure the Jacobian is 1 by using a restricted family of transformations, under which the value of \mathbf{l}_t^k depends only on the current value, $\boldsymbol{\eta}_t$ via simple addition, and on past values of $\boldsymbol{\eta}$ via an arbitrary function,

$$\mathbf{l}_t^{k+1} = \boldsymbol{\eta}_t + \mathbf{f}^{k+1}(\mathbf{x}, \boldsymbol{\eta}_{1:t-1}). \quad (12)$$

And we know that our fixed-point iterations indeed lie in this family of functions, as Eq. (10) mirrors the above definition, with,

$$\mathbf{f}^{k+1}(\mathbf{x}, \boldsymbol{\eta}_{1:t-1}) = \mathbf{v}_t(\mathbf{x}, \boldsymbol{\sigma}(\beta \mathbf{l}_{1:t-1}^k)) \quad (13)$$

where $\mathbf{l}_{1:t-1}^k$ indeed depends only on $\boldsymbol{\eta}_{1:t-1}$ (see Eq. 10).

3.4 Issues with continuous model

Using continuous latent variables might seem appealing, as we can not only compute the exact approximate posterior probability, even when the fixed point iterations have not converged, but the normalizing flow framework gives us considerable flexibility in choosing \mathbf{f}^{k+1} (i.e. we are not restricted to normalizing flows which have an interpretation as a discrete dynamical system). However, moving from a discrete latent variable (Eq. 3 with $u = 0$, and thus $\sigma(u) = 0.5$) to a continuous latent variable (Eq. 4) introduces scope for considerable mismatch between the true and approximate posterior (Fig. 1B), which simply is not possible for the binary model where the posterior remains Bernoulli. This mismatch between the approximate and true posterior implies a discrepancy between the ELBO and the true model evidence (Fig. 1C), and this discrepancy grows as the evidence in favour of $\bar{z} = 0$ or $\bar{z} = 1$ increases. This mismatch has a potentially large magnitude (compare the scale on Fig. 1B to that in Fig. 2F), and thus can dramatically modify the variational inference objective function, introducing the possibility for biased inferences. In fact, this is exactly what we see (Fig. 1BD), with the approximate posterior underestimating the evidence in favour of $\bar{z} = 0$ or $\bar{z} = 1$.

Further, the Logistic approximate posterior can introduce additional biases when optimized in combination with a relaxed binary variables (Figs. 1B–D use a hard-thresholding or equivalently, an infinite inverse-temperature). In particular, we consider a Gaussian likelihood, $P(x = 0.5 | \hat{z}) = \mathcal{N}(x = 0.5; \hat{z}, \sigma)$, and thus, as σ decreases, there is increasing evidence that \hat{z} is close to 0.5. Under the true model, with a hard-threshold, this can never be achieved: \bar{z} must be either 0 or 1. However, if we combine a relaxation (Eq. 4) with a Logistic approximate posterior, it is possible to reduce the variance of the logistic sufficiently such that the relaxed Bernoulli variable, \hat{z} , is indeed close to 0.5. To quantify this effect, we consider how the standard deviation of \hat{z} (which should be 0.5 under the true posterior), varies as we change the standard deviation of the likelihood, σ (Fig. 1E).

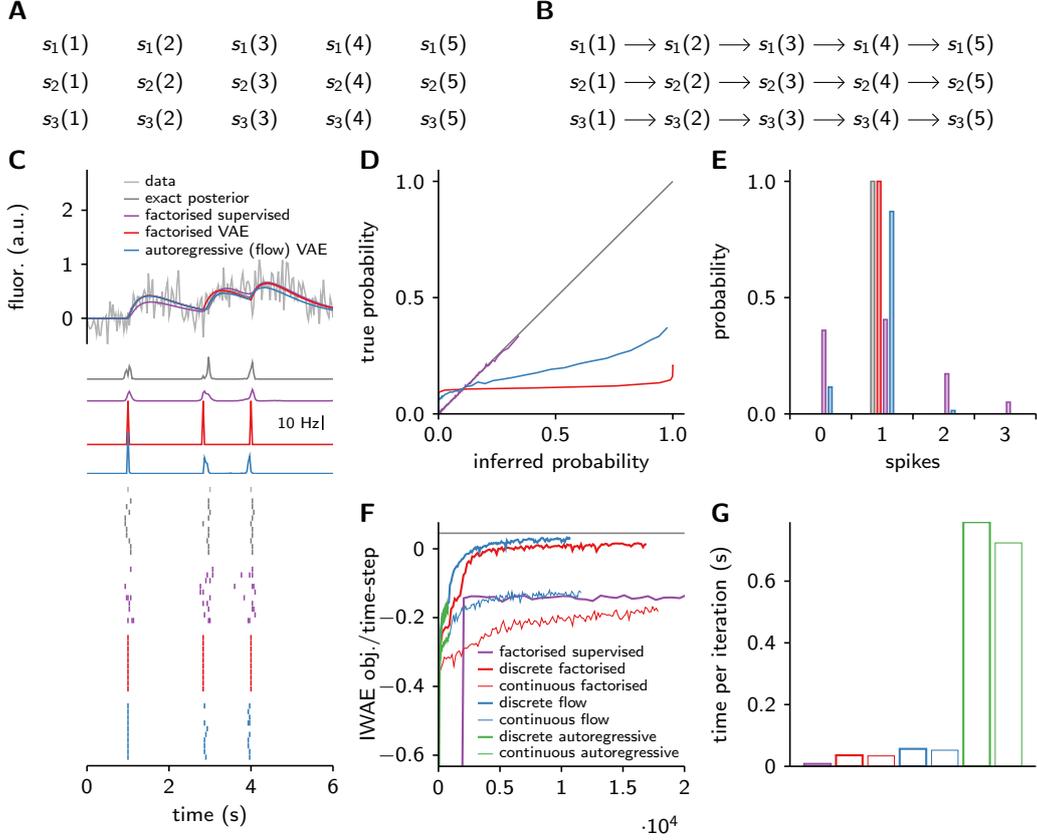


Figure 2: Correlated approximate posteriors in a single-channel (neuron) model. **A** Factorised generative model. **B** Correlated approximate posterior, with an autoregressive temporal structure, but without correlations between cells. **C** Example observed fluorescence trace and average reconstructions (top), inferred rates (middle) and inferred spiking (bottom), for the true posterior (dark gray), then models with factorised posteriors trained using VAE (red) and supervised (purple) procedures, and finally our new discrete flow trained using VAE (blue). **D** The true marginal probability of there being a spike in simulated data, against inferred probability. The optimal is unity (dark-gray). **E** The number of inferred spikes, given only one spike in the underlying data. **F** The time course of the VAE objective under the different models, and the highest possible value for the objective, estimated using importance sampling. **G** The time required for a single iteration of the algorithm, for the different variants in **F**.

3.5 Experiments

One case where binary latent variables are essential is that of calcium spike deconvolution: inferring latent binary variables representing the presence or absence of a neural spike in a small time bin, based on noisy optical observations (Fig. 2C).

We take the binary variables, z_{it} as representing whether neuron i spiked in time-bin t , and as such, we can interpret u_{it} as the corresponding synaptic input (in contrast, as \mathbf{v}_t is not part of the generative model, it does not have a specific biological interpretation). For the generative model, we take a weighted sum of inputs from past spikes, filtered by a temporal kernel κ^u and κ^v , which is mirrored by the recognition model, to ensure that the recognition model can capture any prior-induced statistical structure,

$$\mathbf{u}_t = \mathbf{b}^u + \mathbf{W}^u \sum_{t'=1}^{\tau} \kappa_{t'}^u \odot \mathbf{z}_{t-t'} \quad \mathbf{v}_t = \mathbf{b}^v(\mathbf{x}) + \mathbf{W}^v \sum_{t'=1}^{\tau} \kappa_{t'}^v \odot \mathbf{z}_{t-t'}. \quad (14)$$

where \odot is the Hadamard product, so $\mathbf{r} = \mathbf{g} \odot \mathbf{h}$ implies $r_i = g_i h_i$.

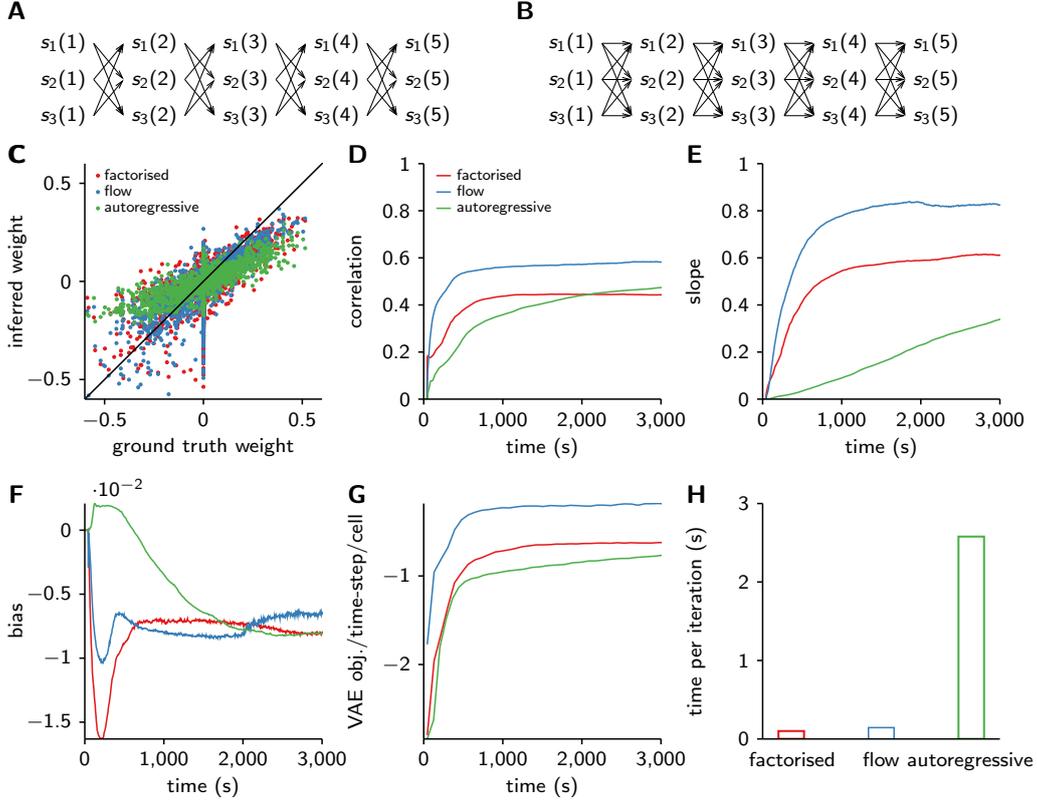


Figure 3: Correlated approximate posteriors to estimate the connectivity between multiple neurons. **A** Autoregressive generative model, incorporating the effects of synaptic connectivity (note the lack of self-connections). **B** Correlated autoregressive approximate posterior. **C** The inferred and ground truth weights using factorised, flow and autoregressive posteriors. **DEF** The correlation (**D**), slope (**E**) and bias (**F**) for the points in **C**, plotted across training time. **G** The time course of the VAE objective under the different models. **H** The time required for a single training iteration of the algorithm, for the different models.

3.5.1 Experiment 1: Single-channel

We considered a single neuron, whose spiking was IID (Poisson) with firing rate 0.25 Hz (Fig. 2A). Fluorescence data was simulated by convolving spikes with a double-exponential temporal kernel with rise time 0.3 s and decay time 1 s and adding noise with standard deviation $e^{-1.5}$. We learned the recognition model, which consisted of two components. The first is a neural network mapping data to spike inferences, $\mathbf{b}^v(\mathbf{x})$, which consisted of two hidden layers, with 20 units per cell per time bin, where the first layer takes input from 200 time points from a single cell’s fluorescence trace, and the second layer takes 5 time points from the previous hidden layer, and we use Elu nonlinearities [14] (for further details see [11, 5]). The second is the recognition temporal kernel κ^v , which captures the anticorrelations induced by explaining away (Fig. 2BC).

All strategies, including factorised and autoregressive (flow) VAE’s, and supervised training give roughly similar reconstructions (Fig. 2C top). Thus, to understand how the autoregressive posterior is superior, we need to look in more depth at the posteriors themselves. In particular, the factorised VAE had very narrow posteriors, spuriously indicating a very high degree of certainty in the spike timing (Fig. 2C middle), whereas both the supervised and autoregressive VAE and also the true posterior (estimate by importance sampling) indicate a higher level of temporal uncertainty. These differences are even more evident if we consider spike trains sampled from the approximate posterior (Fig. 2C bottom), or if we consider calibration: the probability of there actually being a spike in the underlying data, when the inference method under consideration indicates a particular probability of spiking (Fig. 2D). However, the sampled spike trains indicate another issue: while the true and VAE posteriors generally have one spike corresponding to each ground-truth spike, supervised training produces

considerable uncertainty about the spike-count (Fig. 2E). Thus, the VAE with an autoregressive (flow) posterior combines the best of both worlds: achieving reasonable timing uncertainty (unlike the factorised VAE), whilst achieving reasonable spike counts (unlike supervised training).

As such, the autoregressive VAE performs more effectively than the factorised methods considering the IWAE objective (with 10 samples) [15], under which the models are trained (Fig. 2F). As expected given Sec. 3.4, to get good performance, it is important to compute probabilities by putting the relaxed variables into the discrete pmf (binary, as opposed to logistic), and to use the parallel fixed point iteration based flow, as opposed to the autoregressive distribution (note however, that we did not thoroughly explore the space of normalizing flows). The considerable differences in speed between the approximate parallel fixed point iterations, and the exact autoregressive computation arises from an order-of-magnitude difference in the time required for an individual iteration (Fig. 2G).

3.5.2 Experiment 2: Multi-channel

A second problem is inferring synaptic connectivity between cells, based on noisy observations. Here, we considered a network of 100 cells with no self-connectivity, and with weights, \mathbf{W}^u that are sparse (probability of 0.1 of being non-zero), with the non-zero weights drawn from a Gaussian with standard deviation 5 (Fig. 3A), and with a 200 ms temporal decay. We used an autoregressive, fully connected recognition model (Fig. 3B). We use the same parameters as in the above simulation, except that we use a somewhat more realistic fluorescence rise-time of 100 ms (previously, we used 300 ms so as to highlight uncertainty in timing).

We inferred weights under three methods: a factorised VAE, and an autoregressive posterior where we use the fast-flow like sampling procedure (flow), and where we use the slow sequential sampling procedure (autoregressive) (Fig. 3C). The flow posterior gave a considerable advantage over the other methods in terms of the correlation between ground truth and inferred weights (Fig. 3D), and in terms of the slope (Fig. 3E), indicating a reduction in the bias towards underestimating the magnitude of weights, while the bias (i.e. the additive offset in Fig. 3C) remained small for all the methods. As such, the autoregressive posterior with flow-based sampling increased the ELBO considerably over the factorised model, or the autoregressive model with the slow sequential sampling, (Fig. 3G), and these differences again arise because of large differences in the time required for a single training iteration (Fig. 3H).

4 Discussion

We have described an approach to sampling from a discrete autoregressive distribution using a parallel, flow-like procedure, derived by considering a fixed-point iteration that converges to a sample from the underlying autoregressive process. We applied this sampling procedure to autoregressive approximate posteriors in variational inference, in order to sample rapidly from the approximate posterior, and hence to speed up training. This allowed us to rapidly learn autoregressive posteriors in the context of neural data analysis, allowing us to realise the benefits of autoregressive approximate posteriors for single and multi cell data in reasonable timescales.

It is important to remember that while we can sample using K fixed-point iterations, we can only evaluate the probability of a sample once it has converged. This mismatch introduces a level of approximation in addition to those that are typical when relaxing discrete distributions [13, 12], but we can deal with the additional approximation error in the same way: by evaluating the model using samples drawn from the underlying discrete, autoregressive approximate posterior.

Finally, our work suggests two directions that may be interesting to pursue. First, while one straightforwardly use normalizing flows to define approximate posteriors for latent dynamical systems with continuous latent variables, it may be difficult to define approximate posteriors that are well suited to the underlying dynamical process. In this context, our procedure of using fixed-point iterations may be a useful starting point. Second, we showed that while it may be possible to convert a discrete latent variable model to an equivalent model with continuous latents, this typically introduces considerable scope for mismatch between the prior and approximate posterior. However, the actual approximate posterior is relatively simple, a mixture of truncated Logistics, and as such, it may be possible to design approximate posteriors that more closely match the true posterior, though it is less clear how such distributions might be embedded within a normalizing flow.

References

- [1] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *ICLR*, 2014.
- [2] D. J. Rezende, S. Mohamed, and D. Wierstra, “Stochastic backpropagation and approximate inference in deep generative models,” *ICML*, 2014.
- [3] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural networks,” *arXiv preprint arXiv:1505.05424*, 2015.
- [4] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “beta-vae: Learning basic visual concepts with a constrained variational framework,” *ICLR*, 2017.
- [5] L. Aitchison, L. Russell, A. M. Packer, J. Yan, P. Castonguay, M. Häusser, and S. C. Turaga, “Model-based Bayesian inference of neural activity and connectivity from all-optical interrogation of a neural circuit,” in *Advances in Neural Information Processing Systems*, pp. 3486–3495, 2017.
- [6] J. Pearl, “Embracing causality in default reasoning,” *Artificial Intelligence*, vol. 35, pp. 259–271, 1988.
- [7] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [8] D. Rezende and S. Mohamed, “Variational inference with normalizing flows,” in *International Conference on Machine Learning*, pp. 1530–1538, 2015.
- [9] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, “Improved variational inference with inverse autoregressive flow,” in *Advances in Neural Information Processing Systems*, pp. 4743–4751, 2016.
- [10] J. Friedrich, P. Zhou, and L. Paninski, “Fast online deconvolution of calcium imaging data,” *PLoS Computational Biology*, vol. 13, p. e1005423, 2017.
- [11] A. Speiser, J. Yan, E. W. Archer, L. Buesing, S. C. Turaga, and J. H. Macke, “Fast amortized inference of neural activity from calcium imaging data with variational autoencoders,” in *Advances in Neural Information Processing Systems*, pp. 4024–4034, 2017.
- [12] C. J. Maddison, A. Mnih, and Y. W. Teh, “The concrete distribution: A continuous relaxation of discrete random variables,” *arXiv preprint arXiv:1611.00712*, 2016.
- [13] E. Jang, S. Gu, and B. Poole, “Categorical reparameterization with Gumbel-Softmax,” *arXiv preprint arXiv:1611.01144*, 2016.
- [14] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *arXiv preprint arXiv:1511.07289*, 2015.
- [15] Y. Burda, R. Grosse, and R. Salakhutdinov, “Importance weighted autoencoders,” *arXiv preprint arXiv:1509.00519*, 2015.