
Composing graphical models with neural networks for structured representations and fast inference

Matthew James Johnson
Harvard University
`mattjj@seas.harvard.edu`

David Duvenaud
Harvard University
`dduvenaud@seas.harvard.edu`

Alexander B. Wiltschko
Harvard University, Twitter
`awiltsch@fas.harvard.edu`

Sandeep R. Datta
Harvard Medical School
`srdatta@hms.harvard.edu`

Ryan P. Adams
Harvard University, Twitter
`rpa@seas.harvard.edu`

Abstract

We propose a general modeling and inference framework that combines the complementary strengths of probabilistic graphical models and deep learning methods. Our model family composes latent graphical models with neural network observation likelihoods. For inference, we use recognition networks to produce local evidence potentials, then combine them with the model distribution using efficient message-passing algorithms. All components are trained simultaneously with a single stochastic variational inference objective. We illustrate this framework by automatically segmenting and categorizing mouse behavior from raw depth video, and demonstrate several other example models.

1 Introduction

Modeling often has two goals: first, to learn a flexible representation of complex high-dimensional data, such as images or speech recordings, and second, to find structure that is interpretable and generalizes to new tasks. Probabilistic graphical models [1, 2] provide many tools to build structured representations, but often make rigid assumptions and may require significant feature engineering. Alternatively, deep learning methods allow flexible data representations to be learned automatically, but may not directly encode interpretable or tractable probabilistic structure. Here we develop a general modeling and inference framework that combines these complementary strengths.

Consider learning a generative model for video of a mouse. Learning interpretable representations for such data, and comparing them as the animal’s genes are edited or its brain chemistry altered, gives useful behavioral phenotyping tools for neuroscience and for high-throughput drug discovery [3]. Even though each image is encoded by hundreds of pixels, the data lie near a low-dimensional nonlinear manifold. A useful generative model must not only learn this manifold but also provide an interpretable representation of the mouse’s behavioral dynamics. A natural representation from ethology [3] is that the mouse’s behavior is divided into brief, reused actions, such as darts, rears, and grooming bouts. Therefore an appropriate model might switch between discrete states, with each state representing the dynamics of a particular action. These two learning tasks — identifying an image manifold and a structured dynamics model — are complementary: we want to learn the image manifold in terms of coordinates in which the structured dynamics fit well. A similar challenge arises in speech [4], where high-dimensional spectrographic data lie near a low-dimensional manifold because they are generated by a physical system with relatively few degrees of freedom [5] but also include the discrete latent dynamical structure of phonemes, words, and grammar [6].

To address these challenges, we propose a new framework to design and learn models that couple nonlinear likelihoods with structured latent variable representations. Our approach uses graphical models for representing structured probability distributions while enabling fast exact inference subroutines, and uses ideas from variational autoencoders [7, 8] for learning not only the nonlinear

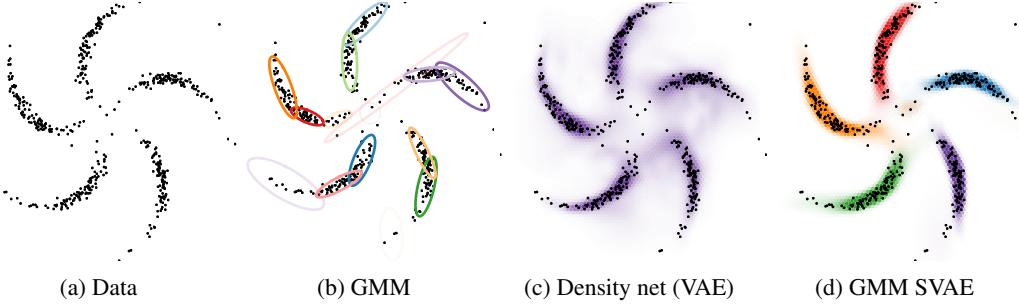


Figure 1: Comparison of generative models fit to spiral cluster data. See Section 2.1.

feature manifold but also bottom-up recognition networks to improve inference. Thus our method enables the combination of flexible deep learning feature models with structured Bayesian (and even nonparametric [9]) priors. Our approach yields a single variational inference objective in which all components of the model are learned simultaneously. Furthermore, we develop a scalable fitting algorithm that combines several advances in efficient inference, including stochastic variational inference [10], graphical model message passing [1], and backpropagation with the reparameterization trick [7]. Thus our algorithm can leverage conjugate exponential family structure where it exists to efficiently compute natural gradients with respect to some variational parameters, enabling effective second-order optimization [11], while using backpropagation to compute gradients with respect to all other parameters. We refer to our general approach as the structured variational autoencoder (SVAE).

2 Latent graphical models with neural net observations

In this paper we propose a broad family of models. Here we develop three specific examples.

2.1 Warped mixtures for arbitrary cluster shapes

One particularly natural structure used frequently in graphical models is the discrete mixture model. By fitting a discrete mixture model to data, we can discover natural clusters or units. These discrete structures are difficult to represent directly in neural network models.

Consider the problem of modeling the data $y = \{y_n\}_{n=1}^N$ shown in Fig. 1a. A standard approach to finding the clusters in data is to fit a Gaussian mixture model (GMM) with a conjugate prior:

$$\pi \sim \text{Dir}(\alpha), \quad (\mu_k, \Sigma_k) \stackrel{\text{iid}}{\sim} \text{NIW}(\lambda), \quad z_n | \pi \stackrel{\text{iid}}{\sim} \pi \quad y_n | z_n, \{(\mu_k, \Sigma_k)\}_{k=1}^K \stackrel{\text{iid}}{\sim} \mathcal{N}(\mu_{z_n}, \Sigma_{z_n}).$$

However, the fit GMM does not represent the natural clustering of the data (Fig. 1b). Its inflexible Gaussian observation model limits its ability to parsimoniously fit the data and their natural semantics.

Instead of using a GMM, a more flexible alternative would be a neural network density model:

$$\gamma \sim p(\gamma) \quad x_n \stackrel{\text{iid}}{\sim} \mathcal{N}(0, I), \quad y_n | x_n, \gamma \stackrel{\text{iid}}{\sim} \mathcal{N}(\mu(x_n; \gamma), \Sigma(x_n; \gamma)), \quad (1)$$

where $\mu(x_n; \gamma)$ and $\Sigma(x_n; \gamma)$ depend on x_n through some smooth parametric function, such as multilayer perceptron (MLP), and where $p(\gamma)$ is a Gaussian prior [12]. This model fits the data density well (Fig. 1c) but does not explicitly represent discrete mixture components, which might provide insights into the data or natural units for generalization. See Fig. 2a for a graphical model.

By composing a latent GMM with nonlinear observations, we can combine the modeling strengths of both [13], learning both discrete clusters along with non-Gaussian cluster shapes:

$$\begin{aligned} \pi &\sim \text{Dir}(\alpha), \quad (\mu_k, \Sigma_k) \stackrel{\text{iid}}{\sim} \text{NIW}(\lambda), \quad \gamma \sim p(\gamma) \\ z_n | \pi &\stackrel{\text{iid}}{\sim} \pi \quad x_n \stackrel{\text{iid}}{\sim} \mathcal{N}(\mu^{(z_n)}, \Sigma^{(z_n)}), \quad y_n | x_n, \gamma \stackrel{\text{iid}}{\sim} \mathcal{N}(\mu(x_n; \gamma), \Sigma(x_n; \gamma)). \end{aligned}$$

This combination of flexibility and structure is shown in Fig. 1d. See Fig. 2b for a graphical model.

2.2 Latent linear dynamical systems for modeling video

Now we consider a harder problem: generatively modeling video. Since a video is a sequence of image frames, a natural place to start is with a model for images. Kingma et al. [7] shows that

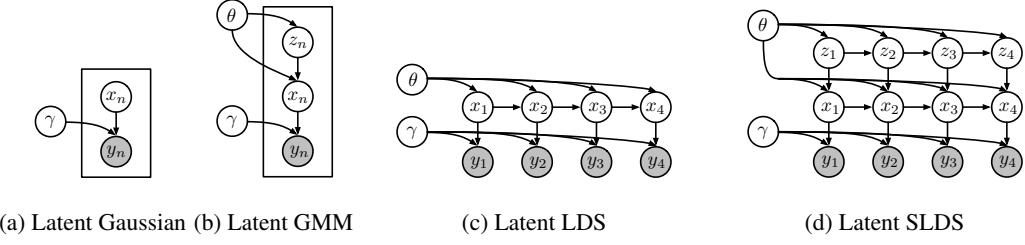


Figure 2: Generative graphical models discussed in Section 2.

the density network of Eq. (2.1) can accurately represent a dataset of high-dimensional images $\{y_n\}_{n=1}^N$ in terms of the low-dimensional latent variables $\{x_n\}_{n=1}^N$, each with independent Gaussian distributions.

To extend this image model into a model for videos, we can introduce dependence through time between the latent Gaussian samples $\{x_n\}_{n=1}^N$. For instance, we can make each latent variable x_{n+1} depend on the previous latent variable x_n through a Gaussian linear dynamical system, writing

$$x_{n+1} = Ax_n + Bu_n, \quad u_n \stackrel{\text{iid}}{\sim} \mathcal{N}(0, I), \quad A, B \in \mathbb{R}^{m \times m},$$

where the matrices A and B have a conjugate prior. This model has low-dimensional latent states and dynamics as well as a rich nonlinear generative model of images. In addition, the timescales of the dynamics are represented directly in the eigenvalue spectrum of A , providing both interpretability and a natural way to encode prior information. See Fig. 2c for a graphical model.

2.3 Latent switching linear dynamical systems for parsing behavior from video

As a final example that combines both time series structure and discrete latent units, consider again the behavioral phenotyping problem described in Section 1. Drawing on graphical modeling tools, we can construct a latent switching linear dynamical system (SLDS) [14] to represent the data in terms of continuous latent states that evolve according to a discrete library of linear dynamics, and drawing on deep learning methods we can generate video frames with a neural network image model.

At each time $n \in \{1, 2, \dots, N\}$ there is a discrete-valued latent state $z_n \in \{1, 2, \dots, K\}$ that evolves according to Markovian dynamics. The discrete state indexes a set of linear dynamical parameters, and the continuous-valued latent state $x_n \in \mathbb{R}^m$ evolves according to the corresponding dynamics,

$$z_{n+1} | z_n, \pi \sim \pi_{z_n}, \quad x_{n+1} = A_{z_n} x_n + B_{z_n} u_n, \quad u_n \stackrel{\text{iid}}{\sim} \mathcal{N}(0, I),$$

where $\pi = \{\pi_k\}_{k=1}^K$ denotes the Markov transition matrix and $\pi_k \in \mathbb{R}_+^K$ is its k th row. We use the same neural net observation model as in Section 2.2. This SLDS model combines both continuous and discrete latent variables with rich nonlinear observations. See Fig. 2d for a graphical model.

3 Structured mean field inference and recognition networks

Why aren't such rich hybrid models used more frequently? The main difficulty with combining rich latent variable structure and flexible likelihoods is inference. The most efficient inference algorithms used in graphical models, like structured mean field and message passing, depend on conjugate exponential family likelihoods to preserve tractable structure. When the observations are more general, like neural network models, inference must either fall back to general algorithms that do not exploit the model structure or else rely on bespoke algorithms developed for one model at a time.

In this section, we review inference ideas from conjugate exponential family probabilistic graphical models and variational autoencoders, which we combine and generalize in the next section.

3.1 Inference in graphical models with conjugacy structure

Graphical models and exponential families provide many algorithmic tools for efficient inference [15]. Given an exponential family latent variable model, when the observation model is a conjugate

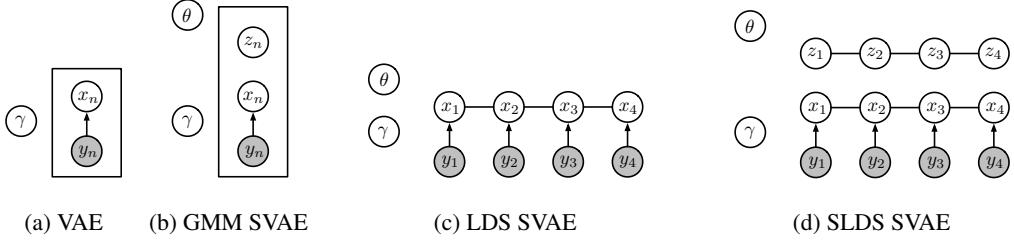


Figure 3: Variational families and recognition networks for the VAE [7] and three SVAE examples.

exponential family, the conditional distributions stay in the same exponential families as in the prior and hence allow for the same efficient inference algorithms.

For example, consider learning a Gaussian linear dynamical system model with linear Gaussian observations. The generative model for latent states $x = \{x_n\}_{n=1}^N$ and observations $y = \{y_n\}_{n=1}^N$ is

$$x_n = Ax_{n-1} + Bu_{n-1}, \quad u_n \stackrel{\text{iid}}{\sim} \mathcal{N}(0, I), \quad y_n = Cx_n + Dv_n, \quad v_n \stackrel{\text{iid}}{\sim} \mathcal{N}(0, I),$$

given parameters $\theta = (A, B, C, D)$ with a conjugate prior $p(\theta)$. To approximate the posterior $p(\theta, x | y)$, consider the mean field family $q(\theta)q(x)$ and the variational inference objective

$$\mathcal{L}[q(\theta)q(x)] = \mathbb{E}_{q(\theta)q(x)} \left[\log \frac{p(\theta)p(x|\theta)p(y|x,\theta)}{q(\theta)q(x)} \right], \quad (2)$$

where we can optimize the variational family $q(\theta)q(x)$ to approximate the posterior $p(\theta, x | y)$ by maximizing Eq. (3.1). Because the observation model $p(y | x, \theta)$ is conjugate to the latent variable model $p(x | \theta)$, for any fixed $q(\theta)$ the optimal factor $q^*(x) \triangleq \arg \max_{q(x)} \mathcal{L}[q(\theta)q(x)]$ is itself a Gaussian linear dynamical system with parameters that are simple functions of the expected statistics of $q(\theta)$ and the data y . As a result, for fixed $q(\theta)$ we can easily compute $q^*(x)$ and use message passing algorithms to perform exact inference in it. However, when the observation model is not conjugate to the latent variable model, these algorithmically exploitable structures break down.

3.2 Recognition networks in variational autoencoders

The variational autoencoder (VAE) [7] handles general non-conjugate observation models by introducing recognition networks. For example, when a Gaussian latent variable model $p(x)$ is paired with a general nonlinear observation model $p(y | x, \gamma)$, the posterior $p(x | y, \gamma)$ is non-Gaussian, and it is difficult to compute an optimal Gaussian approximation. The VAE instead learns to directly output a suboptimal Gaussian factor $q(x | y)$ by fitting a parametric map from data y to a mean and covariance, $\mu(y; \phi)$ and $\Sigma(y; \phi)$, such as an MLP with parameters ϕ . By optimizing over ϕ , the VAE effectively learns how to condition on non-conjugate observations y and produce a good approximating factor.

4 Structured variational autoencoders

We can combine the tractability of conjugate graphical model inference with the flexibility of variational autoencoders. The main idea is to use a conditional random field (CRF) variational family. We learn recognition networks that output conjugate graphical model potentials instead of outputting the complete variational distribution's parameters directly. These potentials are then used in graphical model inference algorithms in place of the non-conjugate observation likelihoods.

The SVAE algorithm computes stochastic gradients of a mean field variational inference objective. It can be viewed as a generalization both of the natural gradient SVI algorithm for conditionally conjugate models [10] and of the AEVB algorithm for variational autoencoders [7]. Intuitively, it proceeds by sampling a data minibatch, applying the recognition model to compute graphical model potentials, and using graphical model inference algorithms to compute the variational factor, combining the evidence from the potentials with the prior structure in the model. This variational factor is then used to compute gradients of the mean field objective. See Fig. 3 for graphical models of the variational families with recognition networks for the models developed in Section 2.

Algorithm 1 Estimate SVAE lower bound and its gradients

Input: Variational parameters $(\eta_\theta, \eta_\gamma, \phi)$, data sample y

function SVAEGRADIENTS($\eta_\theta, \eta_\gamma, \phi, y$)

$$\begin{aligned} \psi &\leftarrow r(y_n; \phi) && \triangleright \text{Get evidence potentials} \\ (\hat{x}, \bar{t}_x, \text{KL}^{\text{local}}) &\leftarrow \text{PGMINFERENCE}(\eta_\theta, \psi) && \triangleright \text{Combine evidence with prior} \\ \hat{\gamma} &\sim q(\gamma) && \triangleright \text{Sample observation parameters} \\ \mathcal{L} &\leftarrow N \log p(y | \hat{x}, \hat{\gamma}) - N \text{KL}^{\text{local}} - \text{KL}(q(\theta)q(\gamma) \| p(\theta)p(\gamma)) && \triangleright \text{Estimate variational bound} \\ \tilde{\nabla}_{\eta_\theta} \mathcal{L} &\leftarrow \eta_\theta^0 - \eta_\theta + N(\bar{t}_x, 1) + N(\nabla_{\eta_x} \log p(y | \hat{x}, \hat{\gamma}), 0) && \triangleright \text{Compute natural gradient} \\ \text{return} &\text{ lower bound } \mathcal{L}, \text{ natural gradient } \tilde{\nabla}_{\eta_\theta} \mathcal{L}, \text{ gradients } \nabla_{\eta_\gamma, \phi} \mathcal{L} \end{aligned}$$

function PGMINFERENCE(η_θ, ψ)

$$q^*(x) \leftarrow \text{OPTIMIZELOCALFACTORS}(\eta_\theta, \psi) \quad \triangleright \text{Fast message-passing inference}$$

return sample $\hat{x} \sim q^*(x)$, statistics $\mathbb{E}_{q^*(x)} t_x(x)$, divergence $\mathbb{E}_{q(\theta)} \text{KL}(q^*(x) \| p(x | \theta))$

In this section, we outline the SVAE model class more formally, write the mean field variational inference objective, and show how to efficiently compute unbiased stochastic estimates of its gradients. The resulting algorithm for computing gradients of the mean field objective, shown in Algorithm 1, is simple and efficient and can be readily applied to a variety of learning problems and graphical model structures. See the supplementals for details and proofs.

4.1 SVAE model class

To set up notation for a general SVAE, we first define a conjugate pair of exponential family densities on global latent variables θ and local latent variables $x = \{x_n\}_{n=1}^N$. Let $p(x | \theta)$ be an exponential family and let $p(\theta)$ be its corresponding natural exponential family conjugate prior, writing

$$\begin{aligned} p(\theta) &= \exp \{ \langle \eta_\theta^0, t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta^0) \}, \\ p(x | \theta) &= \exp \{ \langle \eta_x^0(\theta), t_x(x) \rangle - \log Z_x(\eta_x^0(\theta)) \} = \exp \{ \langle t_\theta(\theta), (t_x(x), 1) \rangle \}, \end{aligned}$$

where we used exponential family conjugacy to write $t_\theta(\theta) = (\eta_x^0(\theta), -\log Z_x(\eta_x^0(\theta)))$. The local latent variables x could have additional structure, like including both discrete and continuous latent variables or tractable graph structure, but here we keep the notation simple.

Next, we define a general likelihood function. Let $p(y | x, \gamma)$ be a general family of densities and let $p(\gamma)$ be an exponential family prior on its parameters. For example, each observation y_n may depend on the latent value x_n through an MLP, as in the density network model of Section 2. This generic non-conjugate observation model provides modeling flexibility, yet the SVAE can still leverage conjugate exponential family structure in inference, as we show next.

4.2 Stochastic variational inference algorithm

Though the general observation model $p(y | x, \gamma)$ means that conjugate updates and natural gradient SVI [10] cannot be directly applied, we show that by generalizing the recognition network idea we can still approximately optimize out the local variational factors leveraging conjugacy structure.

For fixed y , consider the mean field family $q(\theta)q(\gamma)q(x)$ and the variational inference objective

$$\mathcal{L}[q(\theta)q(\gamma)q(x)] \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)} \left[\log \frac{p(\theta)p(\gamma)p(x | \theta)p(y | x, \gamma)}{q(\theta)q(\gamma)q(x)} \right]. \quad (3)$$

Without loss of generality we can take the global factor $q(\theta)$ to be in the same exponential family as the prior $p(\theta)$, and we denote its natural parameters by η_θ . We restrict $q(\gamma)$ to be in the same exponential family as $p(\gamma)$ with natural parameters η_γ . Finally, we restrict $q(x)$ to be in the same exponential family as $p(x | \theta)$, writing its natural parameter as η_x . Using these explicit variational parameters, we write the mean field variational inference objective in Eq. (4.2) as $\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x)$.

To perform efficient optimization of the objective $\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x)$, we consider choosing the variational parameter η_x as a function of the other parameters η_θ and η_γ . One natural choice is to set η_x to be a local partial optimizer of \mathcal{L} . However, without conjugacy structure finding a local partial optimizer may be computationally expensive for general densities $p(y | x, \gamma)$, and in the large data setting this

expensive optimization would have to be performed for each stochastic gradient update. Instead, we choose η_x by optimizing over a surrogate objective $\widehat{\mathcal{L}}$ with conjugacy structure, given by

$$\widehat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) \triangleq \mathbb{E}_{q(\theta)q(x)} \left[\log \frac{p(\theta)p(x|\theta)\exp\{\psi(x; y, \phi)\}}{q(\theta)q(x)} \right], \quad \psi(x; y, \phi) \triangleq \langle r(y; \phi), t_x(x) \rangle,$$

where $\{r(y; \phi)\}_{\phi \in \mathbb{R}^m}$ is some parameterized class of functions that serves as the recognition model. Note that the potentials $\psi(x; y, \phi)$ have a form conjugate to the exponential family $p(x|\theta)$. We define $\eta_x^*(\eta_\theta, \phi)$ to be a local partial optimizer of $\widehat{\mathcal{L}}$ along with the corresponding factor $q^*(x)$,

$$\eta_x^*(\eta_\theta, \phi) \triangleq \arg \min_{\eta_x} \widehat{\mathcal{L}}(\eta_\theta, \eta_x, \phi), \quad q^*(x) = \exp \{ \langle \eta_x^*(\eta_\theta, \phi), t_x(x) \rangle - \log Z_x(\eta_x^*(\eta_\theta, \phi)) \}.$$

As with the variational autoencoder of Section 3.2, the resulting variational factor $q^*(x)$ is suboptimal for the variational objective \mathcal{L} . However, because the surrogate objective has the same form as a variational inference objective for a conjugate observation model, the factor $q^*(x)$ not only is easy to compute but also inherits exponential family and graphical model structure for tractable inference.

Given this choice of $\eta_x^*(\eta_\theta, \phi)$, the SVAE objective is $\mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \triangleq \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi))$. This objective is a lower bound for the variational inference objective Eq. (4.2) in the following sense.

Proposition 4.1 (The SVAE objective lower-bounds the mean field objective)

The SVAE objective function $\mathcal{L}_{\text{SVAE}}$ lower-bounds the mean field objective \mathcal{L} in the sense that

$$\max_{q(x)} \mathcal{L}[q(\theta)q(\gamma)q(x)] \geq \max_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \geq \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \quad \forall \phi \in \mathbb{R}^m,$$

for any parameterized function class $\{r(y; \phi)\}_{\phi \in \mathbb{R}^m}$. Furthermore, if there is some $\phi^* \in \mathbb{R}^m$ such that $\psi(x; y, \phi^*) = \mathbb{E}_{q(\gamma)} \log p(y|x, \gamma)$, then the bound can be made tight in the sense that

$$\max_{q(x)} \mathcal{L}[q(\theta)q(\gamma)q(x)] = \max_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) = \max_{\phi} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi).$$

Thus by using gradient-based optimization to maximize $\mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$ we are maximizing a lower bound on the model log evidence $\log p(y)$. In particular, by optimizing over ϕ we are effectively learning how to condition on observations so as to best approximate the posterior while maintaining conjugacy structure. Furthermore, to provide the best lower bound we may choose the recognition model function class $\{r(y; \phi)\}_{\phi \in \mathbb{R}^m}$ to be as rich as possible.

Choosing $\eta_x^*(\eta_\theta, \phi)$ to be a local partial optimizer of $\widehat{\mathcal{L}}$ provides two computational advantages. First, it allows $\eta_x^*(\eta_\theta, \phi)$ and expectations with respect to $q^*(x)$ to be computed efficiently by exploiting exponential family graphical model structure. Second, it provides a simple expression for an unbiased estimate of the natural gradient with respect to the latent model parameters, as we summarize next.

Proposition 4.2 (Natural gradient of the SVAE objective)

The natural gradient of the SVAE objective $\mathcal{L}_{\text{SVAE}}$ with respect to η_θ is

$$\tilde{\nabla}_{\eta_\theta} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) = (\eta_\theta^0 + \mathbb{E}_{q^*(x)} [(t_x(x), 1)] - \eta_\theta) + (\nabla_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi)), 0). \quad (4)$$

Note that the first term in Eq. (4.2) is the same as the expression for the natural gradient in SVI for conjugate models [10], while a stochastic estimate of the second term is computed automatically as part of the backward pass for computing the gradients with respect to the other parameters, as described next. Thus we have an expression for the natural gradient with respect to the latent model's parameters that is almost as simple as the one for conjugate models and just as easy to compute. Natural gradients are invariant to smooth invertible reparameterizations of the variational family [16, 17] and provide effective second-order optimization updates [18, 11].

The gradients of the objective with respect to the other variational parameters, namely $\nabla_{\eta_\gamma} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$ and $\nabla_\phi \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$, can be computed using the reparameterization trick. To isolate the terms that require the reparameterization trick, we rearrange the objective as

$$\mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) = \mathbb{E}_{q(\gamma)q^*(x)} \log p(y|x, \gamma) - \text{KL}(q(\theta)q^*(x) \| p(\theta, x)) - \text{KL}(q(\gamma) \| p(\gamma)).$$

The KL divergence terms are between members of the same tractable exponential families. An unbiased estimate of the first term can be computed by sampling $\hat{x} \sim q^*(x)$ and $\hat{\gamma} \sim q(\gamma)$ and computing $\nabla_{\eta_\gamma, \phi} \log p(y|\hat{x}, \hat{\gamma})$ with automatic differentiation. Note that the second term in Eq. (4.2) is automatically computed as part of the chain rule in computing $\nabla_\phi \log p(y|\hat{x}, \hat{\gamma})$.

5 Related work

In addition to the papers already referenced, there are several recent papers to which this work is related.

The two papers closest to this work are Krishnan et al. [19] and Archer et al. [20]. In Krishnan et al. [19] the authors consider combining variational autoencoders with continuous state-space models, emphasizing the relationship to linear dynamical systems (also called Kalman filter models). They primarily focus on nonlinear dynamics and an RNN-based variational family, as well as allowing control inputs. However, the approach does not extend to general graphical models or discrete latent variables. It also does not leverage natural gradients or exact inference subroutines.

In Archer et al. [20] the authors also consider the problem of variational inference in general continuous state space models but focus on using a structured Gaussian variational family without considering parameter learning. As with Krishnan et al. [19], this approach does not include discrete latent variables (or any latent variables other than the continuous states). However, the method they develop could be used with an SVAE to handle inference with nonlinear dynamics.

In addition, both Gregor et al. [21] and Chung et al. [22] extend the variational autoencoder framework to sequential models, though they focus on RNNs rather than probabilistic graphical models.

Finally, there is much related work on handling nonconjugate model terms in mean field variational inference. In Khan et al. [23] and Khan et al. [24] the authors present a general scheme that is able to exploit conjugate exponential family structure while also handling arbitrary nonconjugate model factors, including the nonconjugate observation models we consider here. In particular, they propose using a proximal gradient framework and splitting the variational inference objective into a difficult term to be linearized (with respect to mean parameters) and a tractable concave term, so that the resulting proximal gradient update is easy to compute, just like in a fully conjugate model. In Knowles et al. [25], the authors propose performing natural gradient descent with respect to natural parameters on each of the variational factors in turn, and they focus on approximating expectations of nonconjugate energy terms in the objective with model-specific lower-bounds (rather than estimating them with generic Monte Carlo). As in conjugate SVI [10], they observe that, on conjugate factors and with an undamped update (i.e. a unit step size), the natural gradient update reduces to the standard conjugate mean field update.

In contrast to the approaches of Khan et al. [23], Khan et al. [24], and Knowles et al. [25], rather than linearizing intractable terms around the current iterate, in this work we handle intractable terms via recognition networks and amortized inference (and the remaining tractable objective terms are multi-concave in general, analogous to SVI [10]). That is, we use parametric function approximators to learn to condition on evidence in a conjugate form. We expect these approaches to handling nonconjugate objective terms may be complementary, and the best choice may be situation-dependent. For models with local latent variables and datasets where minibatch-based updating is important, using inference networks to compute local variational parameters in a fixed-depth circuit (as in the VAE [7, 8]) or optimizing out the local variational factors using fast conjugate updates (as in conjugate SVI [10]) can be advantageous because in both cases local variational parameters for the entire dataset need not be maintained across updates. The SVAE we propose here is a way to combine the inference network and conjugate SVI approaches.

6 Experiments

We apply the SVAE to both synthetic and real data and demonstrate its ability to learn feature representations and latent structure. Code is available at github.com/mattjj/svae.

6.1 LDS SVAE for modeling synthetic data

Consider a sequence of 1D images representing a dot bouncing from one side of the image to the other, as shown at the top of Fig. 8. We use an LDS SVAE to find a low-dimensional latent state space representation along with a nonlinear image model. The model is able to represent the image accurately and to make long-term predictions with uncertainty. See supplements for details.

This experiment also demonstrates the optimization advantages that can be provided by the natural gradient updates. In Fig. 9a we compare natural gradient updates with standard gradient updates at

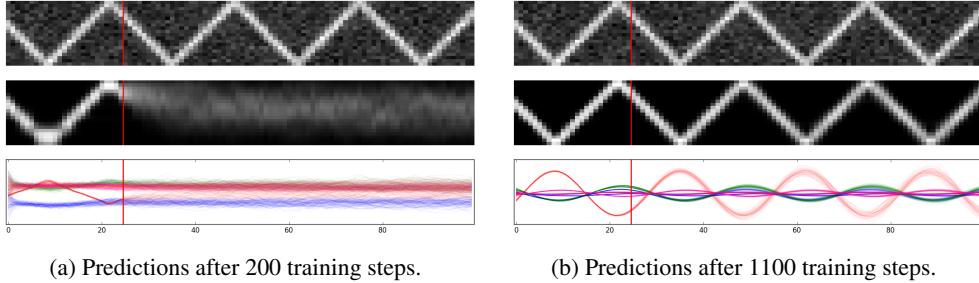


Figure 4: Predictions from an LDS SVAE fit to 1D dot image data at two stages of training. The top panel shows an example sequence with time on the horizontal axis. The middle panel shows the noiseless predictions given data up to the vertical line, while the bottom panel shows the latent states.

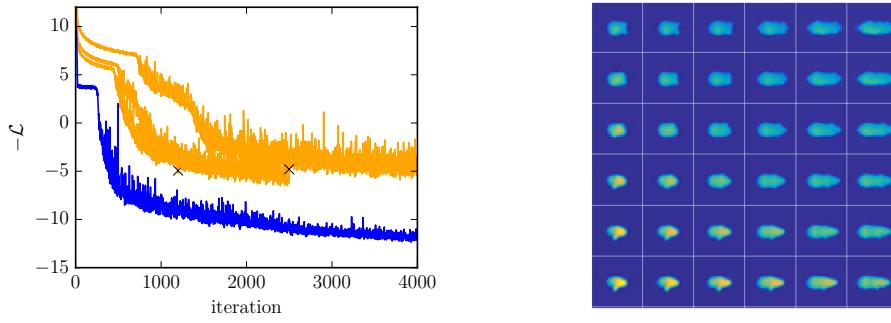


Figure 5: Experimental results from LDS SVAE models on synthetic data and real mouse data.

three different learning rates. The natural gradient algorithm not only learns much faster but also is less dependent on parameterization details: while the natural gradient update used an untuned stepsize of 0.1, the standard gradient dynamics at step sizes of both 0.1 and 0.05 resulted in some matrix parameters to be updated to indefinite values.

6.2 LDS SVAE for modeling video

We also apply an LDS SVAE to model depth video recordings of mouse behavior. We use the dataset from Wiltschko et al. [3] in which a mouse is recorded from above using a Microsoft Kinect. We used a subset consisting of 8 recordings, each of a distinct mouse, 20 minutes long at 30 frames per second, for a total of 288000 video frames downsampled to 30×30 pixels.

We use MLP observation and recognition models with two hidden layers of 200 units each and a 10D latent space. Fig. 9b shows images corresponding to a regular grid on a random 2D subspace of the latent space, illustrating that the learned image manifold accurately captures smooth variation in the mouse’s body pose. Fig. 10 shows predictions from the model paired with real data.

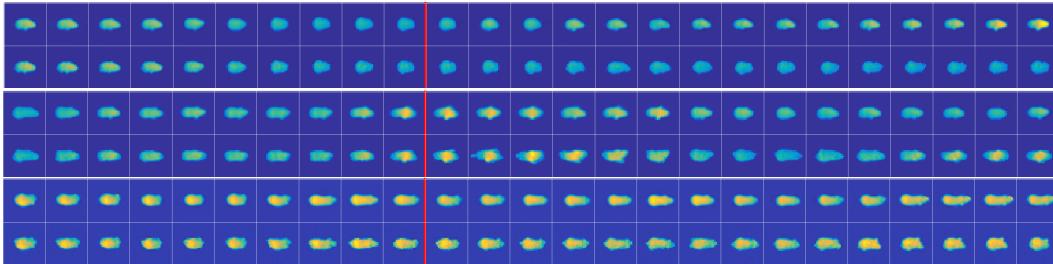


Figure 6: Predictions from an LDS SVAE fit to depth video. In each panel, the top is a sampled prediction and the bottom is real data. The model is conditioned on observations to the left of the line.

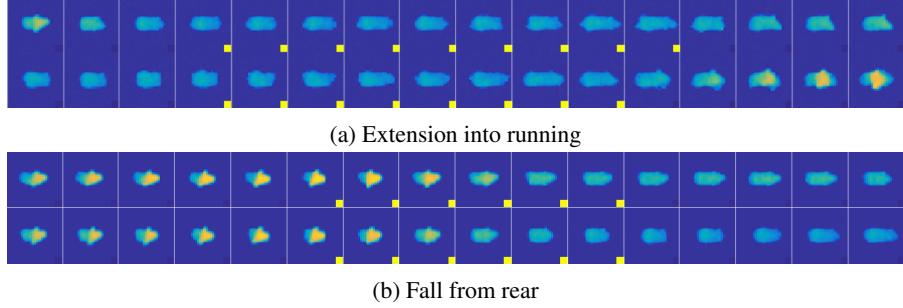


Figure 7: Examples of behavior states inferred from depth video. Each frame sequence is padded on both sides, with a square in the lower-right of a frame depicting when the state is the most probable.

6.3 SLDS SVAE for parsing behavior

Finally, because the LDS SVAE can accurately represent the depth video over short timescales, we apply the latent switching linear dynamical system (SLDS) model to discover the natural units of behavior. Fig. 11 shows some of the discrete states that arise from fitting an SLDS SVAE with 30 discrete states to the depth video data. The discrete states that emerge show a natural clustering of short-timescale patterns into behavioral units. See the supplementals for more.

7 Conclusion

Structured variational autoencoders provide a general framework that combines some of the strengths of probabilistic graphical models and deep learning methods. In particular, they use graphical models both to give models rich latent representations and to enable fast variational inference with CRF-like structured approximating distributions. To complement these structured representations, SVAEs use neural networks to produce not only flexible nonlinear observation models but also fast recognition networks that map observations to conjugate graphical model potentials.

References

- [1] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.
- [2] Kevin P Murphy. *Machine Learning: a Probabilistic Perspective*. MIT Press, 2012.
- [3] Alexander B. Wiltschko, Matthew J. Johnson, Giuliano Iurilli, Ralph E. Peterson, Jesse M. Katon, Stan L. Pashkovski, Victoria E. Abraira, Ryan P. Adams, and Sandeep Robert Datta. “Mapping Sub-Second Structure in Mouse Behavior”. In: *Neuron* 88.6 (2015), pp. 1121–1135.
- [4] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *Signal Processing Magazine, IEEE* 29.6 (2012), pp. 82–97.
- [5] Li Deng. “Computational models for speech production”. In: *Computational Models of Speech Pattern Processing*. Springer, 1999, pp. 199–213.
- [6] Li Deng. “Switching dynamic system models for speech articulation and acoustics”. In: *Mathematical Foundations of Speech and Language Processing*. Springer, 2004, pp. 115–133.
- [7] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *International Conference on Learning Representations* (2014).
- [8] Danilo J Rezende, Shakir Mohamed, and Daan Wierstra. “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”. In: *Proceedings of the 31st International Conference on Machine Learning*. 2014, pp. 1278–1286.
- [9] Matthew J. Johnson and Alan S. Willsky. “Stochastic Variational Inference for Bayesian Time Series Models”. In: *International Conference on Machine Learning*. 2014.
- [10] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. “Stochastic variational inference”. In: *Journal of Machine Learning Research* (2013).
- [11] James Martens. “New insights and perspectives on the natural gradient method”. In: *arXiv preprint arXiv:1412.1193* (2015).
- [12] David J.C. MacKay and Mark N. Gibbs. “Density networks”. In: *Statistics and neural networks: advances at the interface*. Oxford University Press, Oxford (1999), pp. 129–144.
- [13] Tomoharu Iwata, David Duvenaud, and Zoubin Ghahramani. “Warped Mixtures for Nonparametric Cluster Shapes”. In: *Conference on Uncertainty in Artificial Intelligence (UAI)*. 2013, pp. 311–319.
- [14] E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. “Bayesian Nonparametric Inference of Switching Dynamic Linear Models”. In: *IEEE Transactions on Signal Processing* 59.4 (2011).
- [15] Martin J. Wainwright and Michael I. Jordan. “Graphical Models, Exponential Families, and Variational Inference”. In: *Foundations and Trends in Machine Learning* (2008).
- [16] Shun-ichi Amari. “Natural gradient works efficiently in learning”. In: *Neural computation* 10.2 (1998), pp. 251–276.
- [17] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*. American Mathematical Society, 2007.
- [18] James Martens and Roger Grosse. “Optimizing Neural Networks with Kronecker-factored Approximate Curvature”. In: *arXiv preprint arXiv:1503.05671* (2015).
- [19] Rahul G Krishnan, Uri Shalit, and David Sontag. “Deep Kalman Filters”. In: *arXiv preprint arXiv:1511.05121* (2015).
- [20] Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. “Black box variational inference for state space models”. In: *arXiv preprint arXiv:1511.07367* (2015).
- [21] Karol Gregor, Ivo Danihelka, Alex Graves, and Daan Wierstra. “DRAW: A recurrent neural network for image generation”. In: *arXiv preprint arXiv:1502.04623* (2015).
- [22] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. “A recurrent latent variable model for sequential data”. In: *Advances in Neural information processing systems*. 2015, pp. 2962–2970.
- [23] Mohammad E Khan, Pierre Baqué, François Fleuret, and Pascal Fua. “Kullback-Leibler proximal variational inference”. In: *Advances in Neural Information Processing Systems*. 2015, pp. 3402–3410.

- [24] Mohammad Emtiyaz Khan, Reza Babanezhad, Wu Lin, Mark Schmidt, and Masashi Sugiyama. “Faster Stochastic Variational Inference using Proximal-Gradient Methods with General Divergence Functions”. In: *Conference on Uncertainty in Artificial Intelligence (UAI)*. 2016.
- [25] David A Knowles and Tom Minka. “Non-conjugate variational message passing for multinomial and binary regression”. In: *Advances in Neural Information Processing Systems*. 2011, pp. 1701–1709.
- [26] Dimitri P Bertsekas. *Nonlinear programming*. 2nd ed. Athena Scientific, 1999.
- [27] John M. Danskin. *The theory of max-min and its application to weapons allocation problems*. Springer-Verlag, New York, 1967.
- [28] Anthony-V Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*. Academic Press, Inc., 1984.
- [29] J Frederic Bonnans and Alexander Shapiro. *Perturbation Analysis of Optimization Problems*. Springer Science & Business Media, 2000.
- [30] James Martens and Roger Grosse. “Optimizing Neural Networks with Kronecker-factored Approximate Curvature”. In: *Proceedings of the 32nd International Conference on Machine Learning*. 2015.
- [31] David Duvenaud and Ryan P. Adams. “Black-box stochastic variational inference in five lines of Python”. In: *NIPS Workshop on Black-box Learning and Inference* (2015).

A Optimization

In this section we fix our notation for gradients and establish some basic definitions and results that we use in the sequel.

A.1 Gradient notation

We follow the notation in Bertsekas [26, A.5]. In particular, if $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a continuously differentiable function, we define the gradient matrix of f , denoted $\nabla f(x)$, to be the $n \times m$ matrix in which the i th column is the gradient $\nabla f_i(x)$ of f_i , the i th coordinate function of f , for $i = 1, 2, \dots, m$. That is,

$$\nabla f(x) = [\nabla f_1(x) \quad \cdots \quad \nabla f_m(x)].$$

The transpose of ∇f is the Jacobian matrix of f , in which the ij th entry is the function $\partial f_i / \partial x_j$.

If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable with continuously differentiable partial derivatives, then we define the Hessian matrix of f , denoted $\nabla^2 f$, to be the matrix in which the ij th entry is the function $\partial^2 f / \partial x_i \partial x_j$.

Finally, if $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ is a function of (x, y) with $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, we write

$$\begin{aligned}\nabla_x f(x, y) &= \begin{pmatrix} \frac{\partial f(x, y)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x, y)}{\partial x_m} \end{pmatrix}, & \nabla_y f(x, y) &= \begin{pmatrix} \frac{\partial f(x, y)}{\partial y_1} \\ \vdots \\ \frac{\partial f(x, y)}{\partial y_n} \end{pmatrix} \\ \nabla_{xx}^2 f(x, y) &= \left(\frac{\partial^2 f(x, y)}{\partial x_i \partial x_j} \right), & \nabla_{yy}^2 f(x, y) &= \left(\frac{\partial^2 f(x, y)}{\partial y_i \partial y_j} \right), \\ \nabla_{xy}^2 f(x, y) &= \left(\frac{\partial^2 f(x, y)}{\partial x_i \partial y_j} \right).\end{aligned}$$

A.2 Local and partial optimizers

In this section we state the definitions of local partial optimizer and necessary conditions for optimality that we use in the sequel.

Definition A.1 (Partial optimizer, local partial optimizer)

Let $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be an objective function to be maximized. For a fixed $x \in \mathbb{R}^n$, we call a point $y^* \in \mathbb{R}^m$ an unconstrained partial optimizer of f given x if

$$f(x, y) \leq f(x, y^*) \quad \forall y \in \mathbb{R}^m$$

and we call y^* an unconstrained local partial optimizer of f given x if there exists an $\epsilon > 0$ such that

$$f(x, y) \leq f(x, y^*) \quad \forall y \text{ with } \|y - y^*\| < \epsilon,$$

where $\|\cdot\|$ is any vector norm.

Proposition A.2 (Necessary conditions for optimality, Prop. 3.1.1 of Bertsekas [26])

Let $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be continuously differentiable. For fixed $x \in \mathbb{R}^n$ if $y^* \in \mathbb{R}^m$ is an unconstrained local partial optimizer for f given x then

$$\nabla_y f(x, y^*) = 0.$$

If instead x and y are subject to the constraints $h(x, y) = 0$ for some continuously differentiable $h : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ and y^* is a constrained local partial optimizer for f given x with the regularity condition that $\nabla_y h(x, y^*)$ is full rank, then there exists a Lagrange multiplier $\lambda^* \in \mathbb{R}^m$ such that

$$\nabla_y f(x, y^*) + \nabla_y h(x, y^*) \lambda^* = 0,$$

and hence the cost gradient $\nabla_y f(x, y^*)$ is orthogonal to the first-order feasible variations in y given by the null space of $\nabla_y h(x, y^*)^\top$.

Note that the regularity condition on the constraints is not needed if the constraints are linear [26, Prop. 3.3.7].

For a continuously differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, we say x^* is a stationary point of f if $\nabla f(x^*) = 0$. For general unconstrained smooth optimization, the limit points of gradient-based algorithms are guaranteed only to be stationary points of the objective, not necessarily local optima. Block coordinate ascent methods, when available, provide slightly stronger guarantees: not only is every limit point a stationary point of the objective, in addition each coordinate block is a partial optimizer of the objective. Note that the objective functions we consider maximizing in the following are bounded above.

A.3 Partial optimization and the Implicit Function Theorem

Let $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be a scalar-valued objective function of two unconstrained arguments $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, and let $y^* : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be some function that assigns to each $x \in \mathbb{R}^n$ a value $y^*(x) \in \mathbb{R}^m$. Define the composite function $g : \mathbb{R}^n \rightarrow \mathbb{R}$ as

$$g(x) \triangleq f(x, y^*(x))$$

and using the chain rule write its gradient as

$$\nabla g(x) = \nabla_x f(x, y^*(x)) + \nabla y^*(x) \nabla_y f(x, y^*(x)). \quad (5)$$

One choice of the function $y^*(x)$ is to partially optimize f for any fixed value of x . For example, assuming that $\arg \max_y f(x, y)$ is nonempty for every $x \in \mathbb{R}^n$, we could choose y^* to satisfy $y^*(x) \in \arg \max_y f(x, y)$, so that $g(x) = \max_y f(x, y)$.¹ Similarly, if $y^*(x)$ is chosen so that $\nabla_y f(x, y^*(x)) = 0$, which is satisfied when $y^*(x)$ is an unconstrained local partial optimizer for f given x , then the expression in Eq. (A.3) can be simplified as in the following proposition.

Proposition A.3 (Gradients of locally partially optimized objectives)

Let $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be continuously differentiable, let y^* be a local partial optimizer of f given x such that $y^*(x)$ is differentiable, and define $g(x) = f(x, y^*(x))$. Then

$$\nabla g(x) = \nabla_x f(x, y^*(x)).$$

Proof. If y^* is an unconstrained local partial optimizer of f given x then it satisfies $\nabla_y f(x, y^*) = 0$, and if y^* is a regularly-constrained local partial optimizer then the feasible variation $\nabla y^*(x)$ is orthogonal to the cost gradient $\nabla_y f(x, y^*)$. In both cases the second term in the expression for $\nabla g(x)$ in Eq. (A.3) is zero. \square

In general, when $y^*(x)$ is not a stationary point of $f(x, \cdot)$, to evaluate the gradient $\nabla g(x)$ we need to evaluate $\nabla y^*(x)$ in Eq. (A.3). However, this term may be difficult to compute directly. The function $y^*(x)$ may arise implicitly from some system of equations of the form $h(x, y) = 0$ for some continuously differentiable function $h : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$. For example, the value of y may be computed from x and h using a black-box iterative numerical algorithm. However, the Implicit Function Theorem provides another means to compute $\nabla y^*(x)$ using only the derivatives of h and the value of $y^*(x)$.

Proposition A.4 (Implicit Function Theorem, Prop. A.25 of Bertsekas [26])

Let $h : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ be a function and $\bar{x} \in \mathbb{R}^n$ and $\bar{y} \in \mathbb{R}^m$ be points such that

1. $h(\bar{x}, \bar{y}) = 0$
2. h is continuous and has a continuous nonsingular gradient matrix $\nabla_y h(x, y)$ in an open set containing (\bar{x}, \bar{y}) .

Then there exist open sets $S_{\bar{x}} \subseteq \mathbb{R}^n$ and $S_{\bar{y}} \subseteq \mathbb{R}^m$ containing \bar{x} and \bar{y} , respectively, and a continuous function $y^* : S_{\bar{x}} \rightarrow S_{\bar{y}}$ such that $\bar{y} = y^*(\bar{x})$ and $h(x, y^*(x)) = 0$ for all $x \in S_{\bar{x}}$. The function y^* is

¹For a discussion of differentiability issues when there is more than one optimizer, i.e. when $\arg \max_y f(x, y)$ has more than one element, see Danskin [27], Fiacco [28, Section 2.4], and Bonnans et al. [29, Chapter 4]. Here we only consider the sensitivity of local stationary points and assume differentiability almost everywhere.

unique in the sense that if $x \in S_{\bar{x}}$, $y \in S_{\bar{y}}$, and $h(x, y) = 0$, then $y = y^*(x)$. Furthermore, if for some $p > 0$, h is p times continuously differentiable, the same is true for y^* , and we have

$$\nabla y^*(x) = -\nabla_x h(x, y^*(x)) (\nabla_y h(x, y^*(x)))^{-1}, \quad \forall x \in S_{\bar{x}}.$$

As a special case, the equations $h(x, y) = 0$ may be the first-order stationary conditions of another unconstrained optimization problem. That is, the value of y may be chosen by locally partially optimizing the value of $u(x, y)$ for a function $u : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ with no constraints on y , leading to the following corollary.

Corollary A.5 (Implicit Function Theorem for optimization subroutines)

Let $u : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be a twice continuously differentiable function such that the choice $h = \nabla_y u$ satisfies the hypotheses of Proposition A.4 at some point (\bar{x}, \bar{y}) , and define y^* as in Proposition A.4. Then we have

$$\nabla y^*(x) = -\nabla_{xy}^2 u(x, y^*(x)) (\nabla_{yy}^2 u(x, y^*(x)))^{-1}, \quad \forall x \in S_{\bar{x}}.$$

B Exponential families

In this section we set up notation for exponential families and outline some basic results. Throughout this section we take all densities to be absolutely continuous with respect to the appropriate Lebesgue measure (when the underlying set \mathcal{X} is Euclidean space) or counting measure (when \mathcal{X} is discrete), and denote the Borel σ -algebra of a set \mathcal{X} as $\mathcal{B}(\mathcal{X})$ (generated by Euclidean and discrete topologies, respectively). We assume measurability of all functions as necessary.

Given a statistic function $t_x : \mathcal{X} \rightarrow \mathbb{R}^n$ and a base measure $\nu_{\mathcal{X}}$, we can define an exponential family of probability densities on \mathcal{X} relative to $\nu_{\mathcal{X}}$ and indexed by natural parameter $\eta_x \in \mathbb{R}^n$ by

$$p(x | \eta_x) \propto \exp \{ \langle \eta_x, t_x(x) \rangle \}, \quad \forall \eta_x \in \mathbb{R}^n,$$

where $\langle \cdot, \cdot \rangle$ is the standard inner product on \mathbb{R}^n . We also define the partition function as

$$Z_x(\eta_x) \triangleq \int \exp \{ \langle \eta_x, t_x(x) \rangle \} \nu_{\mathcal{X}}(dx)$$

and define $H \subseteq \mathbb{R}^n$ to be the set of all normalizable natural parameters,

$$H \triangleq \{ \eta \in \mathbb{R}^n : Z_x(\eta) < \infty \}.$$

We can write the normalized probability density as

$$p(x | \eta) = \exp \{ \langle \eta_x, t_x(x) \rangle - \log Z_x(\eta_x) \}. \quad (6)$$

We say that an exponential family is *regular* if H is open, and *minimal* if there is no $\eta \in \mathbb{R}^n \setminus \{0\}$ such that $\langle \eta, t_x(x) \rangle = 0$ ($\nu_{\mathcal{X}}$ -a.e.). We assume all families are regular and minimal.² Finally, when we parameterize the family with some other coordinates θ , we write the natural parameter as a continuous function $\eta_x(\theta)$ and write the density as

$$p(x | \theta) = \exp \{ \langle \eta_x(\theta), t_x(x) \rangle - \log Z_x(\eta_x(\theta)) \}$$

and take $\Theta = \eta_x^{-1}(H)$ to be the open set of parameters that correspond to normalizable densities. We summarize this notation in the following definition.

Definition B.1 (Exponential family of densities)

Given a measure space $(\mathcal{X}, \mathcal{B}(\mathcal{X}), \nu_{\mathcal{X}})$, a statistic function $t_x : \mathcal{X} \rightarrow \mathbb{R}^n$, and a natural parameter function $\eta_x : \Theta \rightarrow \mathbb{R}^n$, the corresponding exponential family of densities relative to $\nu_{\mathcal{X}}$ is

$$p(x | \theta) = \exp \{ \langle \eta_x(\theta), t_x(x) \rangle - \log Z_x(\eta_x(\theta)) \},$$

where

$$\log Z_x(\eta_x) \triangleq \log \int \exp \{ \langle \eta_x, t_x(x) \rangle \} \nu_{\mathcal{X}}(dx)$$

is the log partition function.

²Families that are not minimal, like the density of the categorical distribution, can be treated by restricting all algebraic operations to the subspace spanned by the statistic, i.e. to the smallest $V \subset \mathbb{R}^n$ with range $t_x \subseteq V$.

When we write exponential families of densities for different random variables, we change the subscripts on the statistic function, natural parameter function, and log partition function to correspond to the symbol used for the random variable. When the corresponding random variable is clear from context, we drop the subscripts to simplify notation.

The next proposition shows that the log partition function of an exponential family generates cumulants of the statistic.

Proposition B.2 (Gradients of $\log Z$ and expected statistics)

The gradient of the log partition function of an exponential family gives the expected sufficient statistic,

$$\nabla \log Z(\eta) = \mathbb{E}_{p(x|\eta)} [t(x)],$$

where the expectation is over the random variable x with density $p(x|\eta)$. More generally, the moment generating function of $t(x)$ can be written

$$M_{t(x)}(s) \triangleq \mathbb{E}_{p(x|\eta)} \left[e^{\langle s, t(x) \rangle} \right] = e^{\log Z(\eta+s) - \log Z(\eta)}$$

and so derivatives of $\log Z$ give cumulants of $t(x)$, where the first cumulant is the mean and the second and third cumulants are the second and third central moments, respectively.

Given an exponential family of densities on \mathcal{X} as in Definition B.1, we can define a related exponential family of densities on Θ by defining a statistic function $t_\theta(\theta)$ in terms of the functions $\eta_x(\theta)$ and $\log Z_x(\eta_x(\theta))$.

Definition B.3 (Natural exponential family conjugate prior)

Given the exponential family $p(x|\theta)$ of Definition B.1, define the statistic function $t_\theta : \Theta \rightarrow \mathbb{R}^{n+1}$ as the concatenation

$$t_\theta(\theta) \triangleq (\eta_x(\theta), -\log Z_x(\eta_x(\theta))),$$

where the first n coordinates of $t_\theta(\theta)$ are given by $\eta_x(\theta)$ and the last coordinate is given by $-\log Z_x(\eta_x(\theta))$. We call the exponential family with statistic $t_\theta(\theta)$ the natural exponential family conjugate prior to the density $p(x|\theta)$ and write

$$p(\theta) = \exp \{ \langle \eta_\theta, t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta) \}$$

where $\eta_\theta \in \mathbb{R}^{n+1}$ and the density is taken relative to some measure ν_Θ on $(\Theta, \mathcal{B}(\Theta))$.

Notice that using $t_\theta(\theta)$ we can rewrite the original density $p(x|\theta)$ as

$$\begin{aligned} p(x|\theta) &= \exp \{ \langle \eta_x(\theta), t_x(x) \rangle - \log Z_x(\eta_x(\theta)) \} \\ &= \exp \{ \langle t_\theta(\theta), (t_x(x), 1) \rangle \}. \end{aligned}$$

This relationship is useful in Bayesian inference: when the exponential family $p(x|\theta)$ is a likelihood function and the family $p(\theta)$ is used as a prior, the pair enjoy a convenient conjugacy property, as summarized in the next proposition.

Proposition B.4 (Conjugacy)

Let the densities $p(x|\theta)$ and $p(\theta)$ be defined as in Definitions B.1 and B.3, respectively. We have the relations

$$\begin{aligned} p(\theta, x) &= \exp \{ \langle \eta_\theta + (t_x(x), 1), t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta) \} \\ p(\theta|x) &= \exp \{ \langle \eta_\theta + (t_x(x), 1), t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta + (t_x(x), 1)) \} \end{aligned} \tag{7}$$

and hence in particular the posterior $p(\theta|x)$ is in the same exponential family as $p(\theta)$ with the natural parameter $\eta_\theta + (t_x(x), 1)$. Similarly, with multiple likelihood terms $p(x_i|\theta)$ for $i = 1, 2, \dots, N$ we have

$$p(\theta) \prod_{i=1}^N p(x_i|\theta) = \exp \left\{ \langle \eta_\theta + \sum_{i=1}^N (t_x(x_i), 1), t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta) \right\}. \tag{8}$$

Finally, we give a few more exponential family properties that are useful for gradient-based optimization algorithms and variational inference. In particular, we note that the Fisher information matrix of an exponential family can be computed as the Hessian matrix of its log partition function, and that the KL divergence between two members of the same exponential family has a simple expression.

Definition B.5 (Score vector and Fisher information matrix)

Given a family of densities $p(x | \theta)$ indexed by a parameter θ , the score vector $v(x, \theta)$ is the gradient of the log density with respect to the parameter,

$$v(x, \theta) \triangleq \nabla_{\theta} \log p(x | \theta),$$

and the Fisher information matrix for the parameter θ is the covariance of the score,

$$I(\theta) \triangleq \mathbb{E}[v(x, \theta)v(x, \theta)^T],$$

where the expectation is taken over the random variable x with density $p(x | \theta)$, and where we have used the identity $\mathbb{E}[v(x, \theta)] = 0$.

Proposition B.6 (Score and Fisher information for exponential families)

Given an exponential family of densities $p(x | \eta)$ indexed by the natural parameter η , as in Eq. (B), the score with respect to the natural parameter is given by

$$v(x, \eta) = \nabla_{\eta} \log p(x | \eta) = t(x) - \nabla \log Z(\eta)$$

and the Fisher information matrix is given by

$$I(\eta) = \nabla^2 \log Z(\eta).$$

Proposition B.7 (KL divergence in an exponential family)

Given an exponential family of densities $p(x | \eta)$ indexed by the natural parameter η , as in Eq. (B), and two particular members with natural parameters η_1 and η_2 , respectively, the KL divergence from one to the other is

$$\begin{aligned} \text{KL}(p(x | \eta_1) \| p(x | \eta_2)) &\triangleq \mathbb{E}_{p(x | \eta_1)} \left[\log \frac{p(x | \eta_1)}{p(x | \eta_2)} \right] \\ &= \langle \eta_1 - \eta_2, \nabla \log Z(\eta_1) \rangle - (\log Z(\eta_1) - \log Z(\eta_2)). \end{aligned} \quad (9)$$

C Natural gradient SVI for exponential families

In this section we give a derivation of the natural gradient stochastic variational inference (SVI) method of Hoffman et al. [10] using our notation. We extend the algorithm in Section D.

C.1 SVI objective

Let $p(x, y | \theta)$ be an exponential family and $p(\theta)$ be its corresponding natural exponential family prior as in Definitions B.1 and B.3, writing

$$\begin{aligned} p(\theta) &= \exp \{ \langle \eta_{\theta}^0, t_{\theta}(\theta) \rangle - \log Z_{\theta}(\eta_{\theta}^0) \} \\ p(x, y | \theta) &= \exp \{ \langle \eta_{xy}^0(\theta), t_{xy}(x, y) \rangle - \log Z_{xy}(\eta_{xy}^0(\theta)) \} \\ &= \exp \{ \langle t_{\theta}(\theta), (t_{xy}(x, y), 1) \rangle \} \end{aligned} \quad (10)$$

where we have used $t_{\theta}(\theta) = (\eta_{xy}^0(\theta), -\log Z_{xy}(\eta_{xy}^0(\theta)))$ in Eq. (C.1).

Given a fixed observation y , for any density $q(\theta, x) = q(\theta)q(x)$ we have

$$\begin{aligned} \log p(y) &= \mathbb{E}_{q(\theta)q(x)} \left[\log \frac{p(\theta)p(x, y | \theta)}{q(\theta)q(x)} \right] + \text{KL}(q(\theta)q(x) \| p(\theta, x | y)) \\ &\geq \mathbb{E}_{q(\theta)q(x)} \left[\log \frac{p(\theta)p(x, y | \theta)}{q(\theta)q(x)} \right] \end{aligned}$$

where we have used the fact that the KL divergence is always nonnegative. Therefore to choose $q(\theta)q(x)$ to minimize the KL divergence to the posterior $p(\theta, x | y)$ we define the mean field variational inference objective as

$$\mathcal{L}[q(\theta)q(x)] \triangleq \mathbb{E}_{q(\theta)q(x)} \left[\log \frac{p(\theta)p(x, y | \theta)}{q(\theta)q(x)} \right] \quad (11)$$

and the mean field variational inference problem as

$$\max_{q(\theta)q(x)} \mathcal{L}[q(\theta)q(x)]. \quad (12)$$

The following proposition shows that because of the exponential family conjugacy structure, we can fix the parameterization of $q(\theta)$ and still optimize over all possible densities without loss of generality.

Proposition C.1 (Optimal form of the global variational factor)

Given the mean field optimization problem Eq. (C.1), for any fixed $q(x)$ the optimal factor $q(\theta)$ is determined (ν_Θ -a.e.) by

$$q(\theta) \propto \exp \left\{ \langle \eta_\theta^0 + \mathbb{E}_{q(x)} [(t_{xy}(x, y), 1)], t_\theta(\theta) \rangle \right\}.$$

In particular, the optimal $q(\theta)$ is in the same exponential family as the prior $p(\theta)$.

This proposition follows immediately from a more general lemma, which we reuse in the sequel.

Lemma C.2 (Optimizing a mean field factor)

Let $p(a, b, c)$ be a joint density and let $q(a)$, $q(b)$, and $q(c)$ be mean field factors. Consider the mean field variational inference objective

$$\mathbb{E}_{q(a)q(b)q(c)} \left[\log \frac{p(a, b, c)}{q(a)q(b)q(c)} \right].$$

For fixed $q(a)$ and $q(c)$, the partially optimal factor $q^*(b)$ over all possible densities,

$$q^*(b) \triangleq \arg \max_{q(b)} \mathbb{E}_{q(a)q(b)q(c)} \left[\log \frac{p(a, b, c)}{q(a)q(b)q(c)} \right], \quad (13)$$

is defined (almost everywhere) by

$$q^*(b) \propto \exp \left\{ \mathbb{E}_{q(a)q(c)} \log p(a, b, c) \right\}.$$

In particular, if $p(c|b, a)$ is an exponential family with $p(b|a)$ its natural exponential family conjugate prior, and $\log p(b, c|a)$ is a multilinear polynomial in the statistics $t_b(b)$ and $t_c(c)$, written

$$\begin{aligned} p(b|a) &= \exp \left\{ \langle \eta_b^0(a), t_b(b) \rangle - \log Z_b(\eta_b^0(a)) \right\}, \\ p(c|b, a) &= \exp \left\{ \langle \eta_c^0(b, a), t_c(c) \rangle - \log Z_c(\eta_c^0(b, a)) \right\} \\ &= \exp \left\{ \langle t_b(b), \eta_c^0(a)^\top (t_c(c), 1) \rangle \right\}, \end{aligned}$$

for some matrix $\eta_c^0(a)$, then the optimal factor can be written

$$q^*(b) = \exp \left\{ \langle \eta_b^*, t_b(b) \rangle - \log Z_b(\eta_b^*) \right\}, \quad \eta_b^* \triangleq \mathbb{E}_{q(a)} \eta_b^0(a) + \mathbb{E}_{q(a)q(c)} \eta_c^0(a)^\top (t_c(c), 1).$$

As a special case, when c is conditionally independent of b given a , so that $p(c|b, a) = p(c|b)$, then

$$p(c|b) = \exp \left\{ \langle t_b(b), (t_c(c), 1) \rangle \right\}, \quad \eta_b^* \triangleq \mathbb{E}_{q(a)} \eta_b^0(a) + \mathbb{E}_{q(c)} (t_c(c), 1).$$

Proof. Rewrite the objective in Eq. (C.2), dropping terms that are constant with respect to $q(b)$, as

$$\begin{aligned} \mathbb{E}_{q(a)q(b)q(c)} \left[\log \frac{p(a, b, c)}{q(b)} \right] &= \mathbb{E}_{q(b)} \left[\mathbb{E}_{q(a)q(c)} \log p(a, b, c) - \log q(b) \right] \\ &= \mathbb{E}_{q(b)} \left[\log \exp \mathbb{E}_{q(a)q(c)} \log p(a, b, c) - \log q(b) \right] \\ &= -\mathbb{E}_{q(b)} \left[\frac{q(b)}{\tilde{p}(b)} \right] + \text{const} \\ &= -\text{KL}(q(b) \| \tilde{p}(b)) + \text{const}, \end{aligned}$$

where we have defined a new density $\tilde{p}(b) \propto \exp \left\{ \mathbb{E}_{q(a)q(c)} \log p(a, b, c) \right\}$. We can maximize the objective by setting the KL divergence to zero, choosing $q(b) \propto \exp \left\{ \mathbb{E}_{q(a)q(c)} \log p(a, b, c) \right\}$. The rest follows from plugging in the exponential family densities. \square

Proposition C.1 justifies parameterizing the density $q(\theta)$ with variational natural parameters η_θ as

$$q(\theta) = \exp \left\{ \langle \eta_\theta, t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta) \right\}$$

where the statistic function t_θ and the log partition function $\log Z_\theta$ are the same as in the prior family $p(\theta)$. Using this parameterization, we can define the mean field objective as a function of the parameters η_θ , partially optimizing over $q(x)$,

$$\mathcal{L}(\eta_\theta) \triangleq \max_{q(x)} \mathbb{E}_{q(\theta)q(x)} \left[\log \frac{p(\theta)p(x, y | \theta)}{q(\theta)q(x)} \right]. \quad (14)$$

The partial optimization over $q(x)$ in Eq. (C.1) should be read as choosing $q(x)$ to be a local partial optimizer of Eq. (C.1); in general, it may be intractable to find a global partial optimizer, and the results that follow use only first-order stationary conditions on $q(x)$. We refer to this objective function, where we locally partially optimize the mean field objective Eq. (C.1) over $q(x)$, as the SVI objective.

C.2 Easy natural gradients of the SVI objective

By again leveraging the conjugate exponential family structure, we can write a simple expression for the gradient of the SVI objective, and even for its natural gradient.

Proposition C.3 (Gradient of the SVI objective)

Let the SVI objective $\mathcal{L}(\eta_\theta)$ be defined as in Eq. (C.1). Then the gradient $\nabla \mathcal{L}(\eta_\theta)$ is

$$\nabla \mathcal{L}(\eta_\theta) = (\nabla^2 \log Z_\theta(\eta_\theta)) (\eta_\theta^0 + \mathbb{E}_{q^*(x)} [(t_{xy}(x, y), 1)] - \eta_\theta)$$

where $q^*(x)$ is a local partial optimizer of the mean field objective Eq. (C.1) for fixed global variational parameters η_θ .

Proof. First, note that because $q^*(x)$ is a local partial optimizer for Eq. (C.1) by Proposition A.3, we have

$$\nabla \mathcal{L}(\eta_\theta) = \nabla_{\eta_\theta} \mathbb{E}_{q(\theta)q^*(x)} \left[\log \frac{p(\theta)p(x, y | \theta)}{q(\theta)q^*(x)} \right].$$

Next, we use the conjugate exponential family structure and Proposition B.4, Eq. (B.4), to expand

$$\begin{aligned} \mathbb{E}_{q(\theta)q^*(x)} \left[\log \frac{p(\theta)p(x, y | \theta)}{q(\theta)q^*(x)} \right] &= \langle \eta_\theta^0 + \mathbb{E}_{q^*(x)}(t_{xy}(x, y), 1) - \eta_\theta, \mathbb{E}_{q(\theta)}[t_\theta(\theta)] \rangle \\ &\quad - (\log Z_\theta(\eta_\theta^0) - \log Z_\theta(\eta_\theta)). \end{aligned}$$

Note that we can use Proposition B.2 to replace $\mathbb{E}_{q(\theta)}[t_\theta(\theta)]$ with $\nabla \log Z_\theta(\eta_\theta)$. Differentiating with respect to η_θ and using the product rule, we have

$$\begin{aligned} \nabla \mathcal{L}(\eta_\theta) &= \nabla^2 \log Z_\theta(\eta_\theta) (\eta_\theta^0 + \mathbb{E}_{q^*(x)}(t_{xy}(x, y), 1) - \eta_\theta) \\ &\quad - \nabla \log Z_\theta(\eta_\theta) + \nabla \log Z_\theta(\eta_\theta) \\ &= \nabla^2 \log Z_\theta(\eta_\theta) (\eta_\theta^0 + \mathbb{E}_{q^*(x)}(t_{xy}(x, y), 1) - \eta_\theta). \end{aligned}$$

□

As an immediate result of Proposition C.3, the natural gradient [16] defined by

$$\tilde{\nabla} \mathcal{L}(\eta_\theta) \triangleq (\nabla^2 \log Z_\theta(\eta_\theta))^{-1} \nabla \mathcal{L}(\eta_\theta)$$

has an even simpler expression.

Corollary C.4 (Natural gradient of the SVI objective)

The natural gradient of the SVI objective Eq. (C.1) is

$$\tilde{\nabla} \mathcal{L}(\eta_\theta) = \eta_\theta^0 + \mathbb{E}_{q^*(x)} [(t_{xy}(x, y), 1)] - \eta_\theta.$$

The natural gradient corrects for a kind of curvature in the variational family and is invariant to reparameterization of the family [17]. As a result, natural gradient ascent is effectively a second-order quasi-Newton optimization algorithm, and using natural gradients can greatly accelerate the convergence of gradient-based optimization algorithms [30, 11]. It is a remarkable consequence of the exponential family structure that natural gradients of the partially optimized mean field objective with respect to the global variational parameters can be computed efficiently (without any backward pass as would be required in generic reverse-mode differentiation). Indeed, the exponential family conjugacy structure makes the natural gradient of the SVI objective even easier to compute than the flat gradient.

C.3 Stochastic natural gradients for large datasets

The real utility of natural gradient SVI is in its application to large datasets. Consider the model composed of global latent variables θ , local latent variables $x = \{x_n\}_{n=1}^N$, and data $y = \{y_n\}_{n=1}^N$,

$$p(\theta, x, y) = p(\theta) \prod_{n=1}^N p(x_n, y_n | \theta),$$

where each $p(x_n, y_n | \theta)$ is a copy of the same likelihood function with conjugate prior $p(\theta)$. For fixed observations $y = \{y_n\}_{n=1}^N$, let

$$q(\theta, x) = q(\theta) \prod_{n=1}^N q(x_n)$$

be a variational family to approximate the posterior $p(\theta, x | y)$ and consider the SVI objective given by Eq. (C.1). Using Eq. (B.4) of Proposition B.4, it is straightforward to extend the natural gradient expression in Corollary C.4 to an unbiased Monte Carlo estimate which samples terms in the sum over data points.

Corollary C.5 (Unbiased Monte Carlo estimate of the SVI natural gradient)
Using the model and variational family

$$p(\theta, x, y) = p(\theta) \prod_{n=1}^N p(x_n, y_n | \theta), \quad q(\theta)q(x) = q(\theta) \prod_{n=1}^N q(x_n),$$

where $p(\theta)$ and $p(x_n, y_n | \theta)$ are a conjugate pair of exponential families, define $\mathcal{L}(\eta_\theta)$ as in Eq. (C.1). Let the random index \hat{n} be sampled from the set $\{1, 2, \dots, N\}$ and let $p_n > 0$ be the probability it takes value n . Then

$$\tilde{\nabla} \mathcal{L}(\eta_\theta) = \mathbb{E}_{\hat{n}} \left[\eta_\theta^0 + \frac{1}{p_{\hat{n}}} \mathbb{E}_{q^*(x_{\hat{n}})} [(t_{xy}(x_{\hat{n}}, y_{\hat{n}}), 1)] - \eta_\theta \right],$$

where $q^*(x_{\hat{n}})$ is a local partial optimizer of \mathcal{L} given $q(\theta)$.

Proof. Taking expectation over the index \hat{n} , we have

$$\begin{aligned} \mathbb{E}_{\hat{n}} \left[\frac{1}{p_{\hat{n}}} \mathbb{E}_{q^*(x_{\hat{n}})} [(t_{xy}(x_{\hat{n}}, y_{\hat{n}}), 1)] \right] &= \sum_{n=1}^N \frac{p_n}{p_n} \mathbb{E}_{q^*(x_n)} [(t_{xy}(x_n, y_n), 1)] \\ &= \sum_{n=1}^N \mathbb{E}_{q^*(x_n)} [(t_{xy}(x_n, y_n), 1)]. \end{aligned}$$

The remainder of the proof follows from Proposition B.4 and the same argument as in Proposition C.3. \square

The unbiased stochastic gradient developed in Corollary C.5 can be used in a scalable stochastic gradient ascent algorithm. To simplify notation, in the following sections we drop the notation for multiple likelihood terms $p(x_n, y_n | \theta)$ for $n = 1, 2, \dots, N$ and return to working with a single likelihood term $p(x, y | \theta)$. The extension to multiple likelihood terms is immediate.

C.4 Conditionally conjugate models and block updating

The model classes often considered for natural gradient SVI, and the main model classes we consider here, have additional conjugacy structure in the local latent variables. In this section we introduce notation for this extra structure in terms of the additional local latent variables z and discuss the local block coordinate optimization that is often performed to compute the factor $q^*(z)q^*(x)$ for use in the natural gradient expression.

Let $p(z, x, y | \theta)$ be an exponential family and $p(\theta)$ be its corresponding natural exponential family conjugate prior, writing

$$p(\theta) = \exp \{ \langle \eta_\theta^0, t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta^0) \}, \quad (15)$$

$$\begin{aligned} p(z, x, y | \theta) &= \exp \{ \langle \eta_{zxy}^0(\theta), t_{zxy}(z, x, y) \rangle - \log Z_{zxy}(\eta_{zxy}^0(\theta)) \} \\ &= \exp \{ \langle t_\theta(\theta), (t_{zxy}(z, x, y), 1) \rangle \}, \end{aligned} \quad (16)$$

where we have used $t_\theta(\theta) = (\eta_{zxy}^0(\theta), -\log Z_{zxy}(\eta_{zxy}^0(\theta)))$ in Eq. (C.4). Additionally, let $t_{zxy}(z, x, y)$ be a multilinear polynomial in the statistics functions $t_x(x)$, $t_y(y)$, and $t_z(z)$, let $p(z | \theta)$,

$p(x|z, \theta)$, and $p(y|x, z, \theta) = p(y|x, \theta)$ be exponential families, and let $p(z|\theta)$ be a conjugate prior to $p(x|z, \theta)$ and $p(x|z, \theta)$ be a conjugate prior to $p(y|x, \theta)$, so that

$$p(z|\theta) = \exp \{ \langle \eta_z^0(\theta), t_z(z) \rangle - \log Z_z(\eta_z^0(\theta)) \}, \quad (17)$$

$$\begin{aligned} p(x|z, \theta) &= \exp \{ \langle \eta_x^0(z, \theta), t_x(x) \rangle - \log Z_x(\eta_x^0(z, \theta)) \} \\ &= \exp \{ \langle t_z(z), \eta_x^0(\theta)^T(t_x(x), 1) \rangle \}, \end{aligned} \quad (18)$$

$$\begin{aligned} p(y|x, \theta) &= \exp \{ \langle \eta_y^0(x, \theta), t_y(y) \rangle - \log Z_y(\eta_y^0(x, z, \theta)) \} \\ &= \exp \{ \langle t_x(x), \eta_y^0(\theta)^T(t_y(y), 1) \rangle \}, \end{aligned} \quad (19)$$

for some matrices $\eta_x^0(\theta)$ and $\eta_y^0(\theta)$.

This model class includes many common models, including the latent Dirichlet allocation, switching linear dynamical systems with linear-Gaussian emissions, and mixture models and hidden Markov models with exponential family emissions. The conditionally conjugate structure is both powerful and restrictive: while it potentially limits the expressiveness of the model class, it enables block coordinate optimization with very simple and fast updates, as we show next. When conditionally conjugate structure is not present, these local optimizations can instead be performed with generic gradient-based methods and automatic differentiation [31].

Proposition C.6 (Unconstrained block coordinate ascent on $q(z)$ and $q(x)$)

Let $p(\theta, z, x, y)$ be a model as in Eqs. (C.4)-(C.4), and for fixed data y let $q(\theta)q(z)q(x)$ be a corresponding mean field variational family for approximating the posterior $p(\theta, z, x|y)$, with

$$\begin{aligned} q(\theta) &= \exp \{ \langle \eta_\theta, t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta) \}, \\ q(z) &= \exp \{ \langle \eta_z, t_z(z) \rangle - \log Z_z(\eta_z) \}, \\ q(x) &= \exp \{ \langle \eta_x, t_x(x) \rangle - \log Z_x(\eta_x) \}, \end{aligned}$$

and with the mean field variational inference objective

$$\mathcal{L}[q(\theta)q(z)q(x)] = \mathbb{E}_{q(\theta)q(z)q(x)} \left[\log \frac{p(\theta)p(z|\theta)p(x|z,\theta)p(y|x,z,\theta)}{q(\theta)q(z)q(x)} \right].$$

Fixing the other factors, the partial optimizers $q^*(z)$ and $q^*(x)$ for \mathcal{L} over all possible densities are given by

$$\begin{aligned} q^*(z) &\triangleq \arg \max_{q(z)} \mathcal{L}[q(\theta)q(z)q(x)] = \exp \{ \langle \eta_z^*, t_z(z) \rangle - \log Z_z(\eta_z^*) \}, \\ q^*(x) &\triangleq \arg \max_{q(x)} \mathcal{L}[q(\theta)q(z)q(x)] = \exp \{ \langle \eta_x^*, t_x(x) \rangle - \log Z_x(\eta_x^*) \}, \end{aligned}$$

with

$$\eta_z^* = \mathbb{E}_{q(\theta)} \eta_z^0(\theta) + \mathbb{E}_{q(\theta)q(x)} \eta_x^0(\theta)^T(t_x(x), 1), \quad (20)$$

$$\eta_x^* = \mathbb{E}_{q(\theta)q(z)} \eta_x^0(\theta) t_z(z) + \mathbb{E}_{q(\theta)} \eta_y^0(\theta)^T(t_y(y), 1). \quad (21)$$

Proof. This proposition is a consequence of Lemma C.2 and the conjugacy structure. \square

Proposition C.6 gives an efficient block coordinate ascent algorithm: for fixed η_θ , by alternatively updating η_z and η_x according to Eqs. (C.6)-(C.6) we are guaranteed to converge to a stationary point that is partially optimal in the parameters of each factor. In addition, performing each update requires only computing expected sufficient statistics in the variational factors, which means evaluating $\nabla \log Z_\theta(\eta_\theta)$, $\nabla \log Z_z(\eta_z)$, and $\nabla \log Z_x(\eta_x)$, quantities that be computed anyway in a gradient-based optimization routine. The block coordinate ascent procedure leveraging this conditional conjugacy structure is thus not only efficient but also does not require a choice of step size.

Note in particular that this procedure produces parameters $\eta_z^*(\eta_\theta)$ and $\eta_x^*(\eta_\theta)$ that are partially optimal (and hence stationary) for the objective. That is, defining the parameterized mean field variational inference objective as $L(\eta_\theta, \eta_z, \eta_x) = \mathcal{L}[q(\theta)q(z)q(x)]$, for fixed η_θ the block coordinate ascent procedure has limit points η_z^* and η_x^* that satisfy

$$\nabla_{\eta_z} \mathcal{L}(\eta_\theta, \eta_z^*(\eta_\theta), \eta_x^*(\eta_\theta)) = 0, \quad \nabla_{\eta_x} \mathcal{L}(\eta_\theta, \eta_z^*(\eta_\theta), \eta_x^*(\eta_\theta)) = 0.$$

D The SVAE objective and its gradients

In this section we define the SVAE variational lower bound and show how to efficiently compute unbiased stochastic estimates of its gradients, including an unbiased estimate of the natural gradient with respect to the variational parameters with conjugacy structure. The setup here parallels the setup for natural gradient SVI in Section C, but while SVI is restricted to complete-data conjugate models, here we consider more general likelihood models.

D.1 SVAE objective

Let $p(x | \theta)$ be an exponential family and let $p(\theta)$ be its corresponding natural exponential family conjugate prior, as in Definitions B.1 and B.3, writing

$$p(\theta) = \exp \{ \langle \eta_\theta^0, t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta^0) \}, \quad (22)$$

$$\begin{aligned} p(x | \theta) &= \exp \{ \langle \eta_x^0(\theta), t_x(x) \rangle - \log Z_x(\eta_x^0(\theta)) \} \\ &= \exp \{ \langle t_\theta(\theta), (t_x(x), 1) \rangle \}, \end{aligned} \quad (23)$$

where we have used $t_\theta(\theta) = (\eta_x^0(\theta), -\log Z_x(\eta_x^0(\theta)))$ in Eq. (D.1). Let $p(y | x, \gamma)$ be a general family of densities (not necessarily an exponential family) and let $p(\gamma)$ be an exponential family prior on its parameters of the form

$$p(\gamma) = \exp \{ \langle \eta_\gamma^0, t_\gamma(\gamma) \rangle - \log Z_\gamma(\eta_\gamma^0) \}.$$

For fixed y , consider the mean field family of densities $q(\theta, \gamma, x) = q(\theta)q(\gamma)q(x)$ and the mean field variational inference objective

$$\mathcal{L}[q(\theta)q(\gamma)q(x)] \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)} \left[\log \frac{p(\theta)p(\gamma)p(x | \theta)p(y | x, \gamma)}{q(\theta)q(\gamma)q(x)} \right]. \quad (24)$$

By the same argument as in Proposition C.1, without loss of generality we can take the global factor $q(\theta)$ to be in the same exponential family as the prior $p(\theta)$, and we denote its natural parameters by η_θ , writing

$$q(\theta) = \exp \{ \langle \eta_\theta, t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta) \}.$$

We restrict $q(\gamma)$ to be in the same exponential family as $p(\gamma)$ with natural parameters η_γ , writing

$$q(\gamma) = \exp \{ \langle \eta_\gamma, t_\gamma(\gamma) \rangle - \log Z_\gamma(\eta_\gamma) \}.$$

Finally, we restrict³ $q(x)$ to be in the same exponential family as $p(x | \theta)$, writing its natural parameter as η_x . Using these explicit variational natural parameters, we rewrite the mean field variational inference objective in Eq. (D.1) as

$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)} \left[\log \frac{p(\theta)p(\gamma)p(x | \theta)p(y | x, \gamma)}{q(\theta)q(\gamma)q(x)} \right]. \quad (25)$$

To perform efficient optimization in the objective \mathcal{L} defined in Eq. (D.1), we consider choosing the variational parameter η_x as a function of the other parameters η_θ and η_γ . One natural choice is to set η_x to be a local partial optimizer of \mathcal{L} , as in Section C. However, finding a local partial optimizer may be computationally expensive for general densities $p(y | x, \gamma)$, and in the large data setting this expensive optimization would have to be performed for each stochastic gradient update. Instead, we choose η_x by optimizing over a surrogate objective $\hat{\mathcal{L}}$, which we design using exponential family structure to be both easy to optimize and to share curvature properties with the mean field objective \mathcal{L} . The surrogate objective $\hat{\mathcal{L}}$ is

$$\begin{aligned} \hat{\mathcal{L}}(\eta_\theta, \eta_\gamma, \eta_x, \phi) &\triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)} \left[\log \frac{p(\theta)p(\gamma)p(x | \theta)\exp\{\psi(x; y, \phi)\}}{q(\theta)q(\gamma)q(x)} \right] \\ &= \mathbb{E}_{q(\theta)q(x)} \left[\log \frac{p(\theta)p(x | \theta)\exp\{\psi(x; y, \phi)\}}{q(\theta)q(x)} \right] + \text{const}, \end{aligned} \quad (26)$$

³The parametric form for $q(x)$ need not be restricted a priori, but rather without loss of generality given the surrogate objective Eq. (D.1) and the form of ψ used in Eq. (D.1), the optimal factor $q(x)$ is in the same family as $p(x | \theta)$. We treat it as a restriction here so that we can proceed with more concrete notation.

where the constant does not depend on η_x . We define the function $\psi(x; y, \phi)$ to have a form related to the exponential family $p(x | \theta)$,

$$\psi(x; y, \phi) \triangleq \langle r(y; \phi), t_x(x) \rangle, \quad (27)$$

where $\{r(y; \phi)\}_{\phi \in \mathbb{R}^m}$ is some class of functions parameterized by $\phi \in \mathbb{R}^m$, which we assume only to be continuously differentiable in ϕ . We call $r(y; \phi)$ the *recognition model*. We define $\eta_x^*(\eta_\theta, \phi)$ to be a local partial optimizer of $\hat{\mathcal{L}}$,

$$\eta_x^*(\eta_\theta, \phi) \triangleq \arg \min_{\eta_x} \hat{\mathcal{L}}(\eta_\theta, \eta_\gamma, \eta_x, \phi),$$

where the notation above should be interpreted as choosing $\eta_x^*(\eta_\theta, \phi)$ to be a local argument of maximum. The results to follow rely only on necessary first-order conditions for unconstrained local optimality.

Given this choice of function $\eta_x^*(\eta_\theta, \phi)$, we define the SVAE objective to be

$$\mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \triangleq \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi)), \quad (28)$$

where \mathcal{L} is the mean field variational inference defined in Eq. (D.1), and we define the SVAE optimization problem to be

$$\max_{\eta_\theta, \eta_\gamma, \phi} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi).$$

We summarize these definitions in the following.

Definition D.1 (SVAE objective)

Let \mathcal{L} denote the mean field variational inference objective

$$\mathcal{L}[q(\theta)q(\gamma)q(x)] \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)} \left[\log \frac{p(\theta)p(\gamma)p(x | \theta)p(y | x, \gamma)}{q(\theta)q(\gamma)q(x)} \right], \quad (29)$$

where the densities $p(\theta)$, $p(\gamma)$, and $p(x | \theta)$ are exponential families and $p(\theta)$ is the natural exponential family conjugate prior to $p(x | \theta)$, as in Eqs. (D.1)-(D.1). Given a parameterization of the variational factors as

$$q(\theta) = \exp \{ \langle \eta_\theta, t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta) \}, \quad q(\gamma) = \exp \{ \langle \eta_\gamma, t_\gamma(\gamma) \rangle - \log Z_\gamma(\eta_\gamma) \}, \\ q(x) = \exp \{ \langle \eta_x, t_x(x) \rangle - \log Z_x(\eta_x) \},$$

let $\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x)$ denote the mean field variational inference objective Eq. (D.1) as a function of these variational parameters. We define the SVAE objective as

$$\mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \triangleq \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi)),$$

where $\eta_x^*(\eta_\theta, \phi)$ is defined as a local partial optimizer of the surrogate objective $\hat{\mathcal{L}}$,

$$\eta_x^*(\eta_\theta, \phi) \triangleq \arg \max_{\eta_x} \hat{\mathcal{L}}(\eta_\theta, \eta_x^*(\eta_\theta, \phi), \phi),$$

where the surrogate objective $\hat{\mathcal{L}}$ is defined as

$$\hat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) \triangleq \mathbb{E}_{q(\theta)q(x)} \left[\log \frac{p(\theta)p(x | \theta) \exp\{\psi(x; y, \phi)\}}{q(\theta)q(x)} \right], \\ \psi(x; y, \phi) \triangleq \langle r(y; \phi), t_x(x) \rangle,$$

for some recognition model $r(y; \phi)$ parameterized by $\phi \in \mathbb{R}^m$.

The SVAE objective $\mathcal{L}_{\text{SVAE}}$ is a lower-bound for the partially-optimized mean field variational inference objective in the following sense.

Proposition D.2 (The SVAE objective lower-bounds the mean field objective)

The SVAE objective function $\mathcal{L}_{\text{SVAE}}$ lower-bounds the partially-optimized mean field objective \mathcal{L} in the sense that

$$\max_{q(x)} \mathcal{L}[q(\theta)q(\gamma)q(x)] \geq \max_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \geq \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \quad \forall \phi \in \mathbb{R}^m,$$

for any choice of function class $\{r(y; \phi)\}_{\phi \in \mathbb{R}^m}$ in Eq. (D.1). Furthermore, if there is some $\phi^* \in \mathbb{R}^m$ such that

$$\psi(x; y, \phi^*) = \mathbb{E}_{q(\gamma)} \log p(y | x, \gamma)$$

then the bound can be made tight in the sense that

$$\max_{q(x)} \mathcal{L}[q(\theta)q(\gamma)q(x)] = \max_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) = \max_{\phi} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi).$$

Proof. The inequalities follow from the variational principle and the definition of the SVAE objective $\mathcal{L}_{\text{SVAE}}$. In particular, by Lemma C.2 the optimal factor over all possible densities is given by

$$q^{**}(x) \propto \exp \{ \langle \mathbb{E}_{q(\theta)} \eta_x^0(\theta), t_x(x) \rangle + \mathbb{E}_{q(\gamma)} \log p(y | x, \gamma) \}, \quad (30)$$

while we restrict the factor $q(x)$ to have a particular exponential family form indexed by parameter η_x , namely $q(x) \propto \exp \{ \langle \eta_x, t_x(x) \rangle \}$. In the definition of $\mathcal{L}_{\text{SVAE}}$ we also restrict the parameter η_x to be set to $\eta_x^*(\eta_\theta, \phi)$, a particular function of η_θ and ϕ , rather than setting it to the value that maximizes the mean field objective \mathcal{L} . Finally, equality holds when we can set ϕ to match the optimal η_x and that choice yields the optimal factor given in Eq. (D.1). \square

Proposition D.2 motivates the SVAE optimization problem: by using gradient-based optimization to maximize $\mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$ we are maximizing a lower-bound on the model evidence $\log p(y)$ and correspondingly minimizing the KL divergence from our variational family to the target posterior. Furthermore, it motivates choosing the recognition model function class $\{r(y; \phi)\}_{\phi \in \mathbb{R}^m}$ to be as rich as possible.

As we show in the following, choosing $\eta_x^*(\eta_\theta, \phi)$ to be a local partial optimizer of the surrogate objective $\hat{\mathcal{L}}$ provides two significant computational advantages. First, it allows us to provide a simple expression for an unbiased estimate of the natural gradient $\tilde{\nabla}_{\eta_\theta} \mathcal{L}_{\text{SVAE}}$, as we describe next in Section D.2. Second, it allows $\eta_x^*(\eta_\theta, \phi)$ to be computed efficiently by exploiting exponential family structure, as we show in Section D.4.

D.2 Estimating the natural gradient $\tilde{\nabla}_{\eta_\theta} \mathcal{L}_{\text{SVAE}}$

The definition of η_x^* in terms of the surrogate objective $\hat{\mathcal{L}}$ enables a simple and computationally efficient expression for the natural gradient with respect to the conjugate global variational parameters, $\tilde{\nabla}_{\eta_\theta} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$, as we show in the next proposition.

Proposition D.3 (Natural gradient of the SVAE objective)

The natural gradient of the SVAE objective Eq. (D.1) with respect to the conjugate global variational parameters η_θ is

$$\tilde{\nabla}_{\eta_\theta} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) = (\eta_\theta^0 + \mathbb{E}_{q^*(x)} [(t_x(x), 1)] - \eta_\theta) + (\nabla_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi)), 0)$$

where the first term is the SVI natural gradient from Corollary C.4, using

$$q^*(x) \triangleq \exp \{ \langle \eta_x^*(\eta_\theta, \phi), t_x(x) \rangle - \log Z_x(\eta_x^*(\eta_\theta, \phi)) \},$$

and where a stochastic estimate of the second term is computed as part of the backward pass for the gradient $\nabla_\phi \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi))$.

Proof. First we use the chain rule, analogously to Eq. (A.3), to write the gradient as

$$\begin{aligned} \nabla_{\eta_\theta} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) &= (\nabla^2 \log Z_\theta(\eta_\theta)) (\eta_\theta^0 + \mathbb{E}_{q^*(x)} [(t_{xy}(x, y), 1)] - \eta_\theta) \\ &\quad + (\nabla_{\eta_\theta} \eta_x^*(\eta_\theta, \phi)) (\nabla_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi))), \end{aligned} \quad (31)$$

where the first term is the same as the SVI gradient derived in Proposition C.3. In the case of SVI, the second term is zero because η_x^* is chosen as a partial optimizer of \mathcal{L} , but for the SVAE objective the second term is nonzero in general, and the remainder of this proof amounts to deriving a simple expression for it.

We compute the term $\nabla_{\eta_\theta} \eta_x^*(\eta_\theta, \phi)$ in Eq. (D.2) in terms of the gradients of the surrogate objective $\widehat{\mathcal{L}}$ using the Implicit Function Theorem given in Corollary A.5, which yields

$$\nabla_{\eta_\theta} \eta_x^*(\eta_\theta, \phi) = -\nabla_{\eta_\theta \eta_x}^2 \widehat{\mathcal{L}}(\eta_\theta, \eta_x^*(\eta_\theta, \phi), \phi) \left(\nabla_{\eta_x \eta_x}^2 \widehat{\mathcal{L}}(\eta_\theta, \eta_x^*(\eta_\theta, \phi), \phi) \right)^{-1}. \quad (32)$$

First, we compute the gradient of $\widehat{\mathcal{L}}$ with respect to η_x , writing

$$\begin{aligned} \nabla_{\eta_x} \widehat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) &= \nabla_{\eta_x} \left[\mathbb{E}_{q(\theta)q(x)} \left[\log \frac{p(x | \theta) \exp\{\psi(x; y, \phi)\}}{q(x)} \right] \right] \\ &= \nabla_{\eta_x} \left[\langle \mathbb{E}_{q(\theta)} \eta_x^0(\theta) + r(y; \phi) - \eta_x, \nabla \log Z_x(\eta_x) \rangle + \log Z_x(\eta_x) \right] \\ &= (\nabla^2 \log Z_x(\eta_x)) (\mathbb{E}_{q(\theta)} \eta_x^0(\theta) + r(y; \phi) - \eta_x). \end{aligned} \quad (33)$$

Thus as a consequence of the first-order stationary condition $\nabla_{\eta_x} \widehat{\mathcal{L}}(\eta_\theta, \eta_x^*(\eta_\theta, \phi), \phi) = 0$ and the fact that $\nabla^2 \log Z_x(\eta_x)$ is always positive definite for minimal exponential families, we have

$$\mathbb{E}_{q(\theta)} \eta_x^0(\theta) + r(y; \phi) - \eta_x^*(\eta_\theta, \phi) = 0, \quad (34)$$

which is useful in simplifying the expressions to follow.

Continuing with the calculation of the terms in Eq. (D.2), we compute $\nabla_{\eta_x \eta_x}^2 \widehat{\mathcal{L}}$ by differentiating the expression in Eq. (D.2) again, writing

$$\begin{aligned} \nabla_{\eta_x \eta_x}^2 \widehat{\mathcal{L}}(\eta_\theta, \eta_x^*(\eta_\theta, \phi), \phi) &= -\nabla^2 \log Z_x(\eta_x^*(\eta_\theta, \phi)) \\ &\quad + (\nabla^3 \log Z_x(\eta_x^*(\eta_\theta, \phi))) (\mathbb{E}_{q(\theta)} \eta_x^0(\theta) + r(y; \phi) - \eta_x^*(\eta_\theta, \phi)) \\ &= -\nabla^2 \log Z_x(\eta_x^*(\eta_\theta, \phi)), \end{aligned} \quad (35)$$

where the last line follows from using the first-order stationary condition Eq. (D.2). Next, we compute the other term $\nabla_{\eta_\theta \eta_x}^2 \widehat{\mathcal{L}}$ by differentiating Eq. (D.2) with respect to η_θ to yield

$$\nabla_{\eta_\theta \eta_x}^2 \widehat{\mathcal{L}}(\eta_\theta, \eta_x^*(\eta_\theta, \phi), \phi) = (\nabla^2 \log Z_\theta(\eta_\theta)) \begin{pmatrix} \nabla^2 \log Z_x(\eta_x^*(\eta_\theta, \phi)) \\ 0 \end{pmatrix},$$

where the latter matrix is $\nabla^2 \log Z_x(\eta_x^*(\eta_\theta, \phi))$ padded by a row of zeros.

Plugging these expressions back into Eq. (D.2) and cancelling, we arrive at

$$\nabla_{\eta_\theta} \eta_x^*(\eta_\theta, \phi) = \nabla^2 \log Z_\theta(\eta_\theta) \begin{pmatrix} I \\ 0 \end{pmatrix},$$

and so we have an expression for the gradient of the SVAE objective as

$$\begin{aligned} \nabla_{\eta_\theta} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) &= (\nabla^2 \log Z_\theta(\eta_\theta)) (\eta_\theta^0 + \mathbb{E}_{q^*(x)} [(t_{xy}(x, y), 1)] - \eta_\theta) \\ &\quad + (\nabla^2 \log Z_\theta(\eta_\theta)) (\nabla_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi))), 0. \end{aligned}$$

When we compute the natural gradient, the Fisher information matrix factors on the left of each term cancel, yielding the result in the proposition. \square

As a consequence of Proposition D.3, the SVAE algorithm is almost as simple as the SVI algorithm, which applies only to complete-data conjugate models, yet can handle general likelihood densities $p(y | x, \gamma)$. In particular, to compute the natural gradient $\tilde{\nabla}_{\eta_\theta} \mathcal{L}_{\text{SVAE}}$ we only need to compute the same expected sufficient statistics as in the SVI algorithm plus a correction term that, as we show next, is already estimated via automatic differentiation as part of estimating the gradient $\nabla_\phi \mathcal{L}_{\text{SVAE}}$.

The proof of Proposition D.3 used the necessary condition for unconstrained local optimality to simplify the expression in Eq. (D.2). This simplification does not necessarily hold if η_x is constrained; for example, if the factor $q(x)$ has additional factorization structure that is not present in $p(x | \theta)$, that additional structure can manifest as linear constraints on the natural parameter η_x . In the cases we consider here and in Section D.4 the factorization structure in the variational family matches that in the model and so the stationarity conditions apply. Note also that when $q(x)$ is a Gaussian family with fixed covariance (that is, with sufficient statistics $t_x(x) = x$) the same simplification always applies because third and higher-order cumulants are zero for such families and hence $\nabla^3 \log Z_x(\eta_x) = 0$.

D.3 Estimating the gradients $\nabla_\phi \mathcal{L}_{\text{SVAE}}$ and $\nabla_{\eta_\gamma} \mathcal{L}_{\text{SVAE}}$

To compute an unbiased stochastic estimate of the gradients $\nabla_\phi \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$ and $\nabla_{\eta_\gamma} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$ we use the reparameterization trick [7], which is simply to differentiate a stochastic estimate of the objective $\mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$ as a function of ϕ and η_γ . To isolate the terms that require this sample-based approximation from those that can be computed directly, we rewrite the objective as

$$\mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) = \mathbb{E}_{q(\gamma)q^*(x)} \log p(y | x, \gamma) - \text{KL}(q(\theta)q(\gamma)q^*(x) \| p(\theta, \gamma, x)) \quad (36)$$

where, as before,

$$q^*(x) \triangleq \exp \{ \langle \eta_x^*(\eta_\theta, \phi), t_x(x) \rangle - \log Z_x(\eta_x^*(\eta_\theta, \phi)) \}$$

and so the dependence of the expression in Eq. (D.3) on ϕ is through $\eta_x^*(\eta_\theta, \phi)$.

Only the first term in Eq. (D.3) needs to be estimated with the reparameterization trick. Due to the exponential family structure, the second term in Eq. (D.3) has a simple expression, as we show in the following proposition.

Proposition D.4 (Computing $\text{KL}(q(\theta)q(\gamma)q^*(x) \| p(\theta, \gamma, x))$)

The KL divergence term in the SVAE objective in Eq. (D.3) can be computed as

$$\begin{aligned} \text{KL}(q(\theta)q(\gamma)q^*(x) \| p(\theta, \gamma, x)) &= \text{KL}(q(\theta) \| p(\theta)) + \text{KL}(q(\gamma) \| p(\gamma)) \\ &\quad + \log Z_x(\eta_x^*(\eta_\theta, \phi)) - \langle r(y; \phi), \nabla \log Z_x(\eta_x^*(\eta_\theta, \phi)) \rangle, \end{aligned}$$

where the first two terms can be computed with Eq. (B.7) of Proposition B.7.

Proof. Using basic properties of the KL divergence for factorized densities we have

$$\begin{aligned} \text{KL}(q(\theta)q(\gamma)q^*(x) \| p(\theta, \gamma, x)) &= \text{KL}(q(\theta) \| p(\theta)) + \text{KL}(q(\gamma) \| p(\gamma)) \\ &\quad + \mathbb{E}_{q(\theta)} \text{KL}(q^*(x) \| p(x | \theta)), \end{aligned}$$

and so it remains to compute an expression for the final term. Using Proposition B.7 and the stationarity condition given in Eq. (D.2), we have

$$\begin{aligned} \mathbb{E}_{q(\theta)} \text{KL}(q^*(x) \| p(x | \theta)) &= \langle \mathbb{E}_{q(\theta)} \eta_x^0(\theta) - \eta_x^*(\eta_\theta, \phi), \nabla \log Z_x(\eta_x^*(\eta_\theta, \phi)) \rangle \\ &\quad + \log Z_x(\eta_x^*(\eta_\theta, \phi)) \\ &= \log Z_x(\eta_x^*(\eta_\theta, \phi)) - \langle r(y; \phi), \nabla \log Z_x(\eta_x^*(\eta_\theta, \phi)) \rangle. \end{aligned}$$

Adding these terms yields the desired result. \square

Proposition D.4 gives an explicit formula for computing the KL divergence term involving only the log partition functions of the variational exponential families and their gradients. When the variational family is chosen so that these log partition functions can be computed efficiently, the gradients of this term with respect to ϕ and η_γ can also be computed efficiently by reverse-mode automatic differentiation.

We summarize the results of this subsection in the following proposition.

Proposition D.5 (Estimating $\nabla_\phi \mathcal{L}_{\text{SVAE}}$ and $\nabla_{\eta_\gamma} \mathcal{L}_{\text{SVAE}}$)

Let $\hat{\gamma}(\eta_\gamma) \sim q(\gamma)$ and $\hat{x}(\phi) \sim q^*(x)$ be samples of $q(\gamma)$ and $q^*(x)$, respectively. Unbiased estimates of the gradients $\nabla_\phi \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$ and $\nabla_{\eta_\gamma} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$ are given by

$$\begin{aligned} \nabla_\phi \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) &\approx \nabla_\phi \log p(y | \hat{x}(\phi), \hat{\gamma}(\eta_\gamma)) - \nabla_\phi \text{KL}(q(\theta)q(\gamma)q^*(x) \| p(\theta, \gamma, x)), \\ \nabla_{\eta_\gamma} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) &\approx \nabla_{\eta_\gamma} \log p(y | \hat{x}(\phi), \hat{\gamma}(\eta_\gamma)) - \nabla_{\eta_\gamma} \text{KL}(q(\theta)q(\gamma)q^*(x) \| p(\theta, \gamma, x)). \end{aligned}$$

Both of these gradients can be computed by automatically differentiating the Monte Carlo estimate of $\mathcal{L}_{\text{SVAE}}$ given by

$$\mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \approx \log p(y | \hat{x}(\phi), \hat{\gamma}(\eta_\gamma)) - \text{KL}(q(\theta)q(\gamma)q^*(x) \| p(\theta, \gamma, x))$$

with respect to η_γ and ϕ , respectively, where the second term can be computed via Proposition D.4.

D.4 Partially optimizing $\widehat{\mathcal{L}}$ using conjugacy structure

In Section D.1 we defined the SVAE objective in terms of a function $\eta_x^*(\eta_\theta, \phi)$, which was itself implicitly defined in terms of first-order stationary conditions for an auxiliary objective $\widehat{\mathcal{L}}(\eta_\theta, \eta_x, \phi)$. Here we show how $\widehat{\mathcal{L}}$ admits efficient local partial optimization in the same way as the conditionally conjugate model of Section C.4.

In this section we consider additional structure in the local latent variables. Specifically, as in Section C.4, we introduce to the notation another set of local latent variables z in addition to the local latent variables x . However, unlike Section C.4, we still consider general likelihood families $p(y | x, \gamma)$.

Let $p(z, x | \theta)$ be an exponential family and $p(\theta)$ be its corresponding natural exponential family conjugate prior, writing

$$\begin{aligned} p(\theta) &= \exp \left\{ \langle \eta_\theta^0, t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta^0) \right\}, \\ p(z, x | \theta) &= \exp \left\{ \langle \eta_{zx}^0(\theta), t_{zx}(z, x) \rangle - \log Z_{zx}(\eta_{zx}^0(\theta)) \right\} \\ &= \exp \left\{ \langle t_\theta(\theta), (t_{zx}(z, x), 1) \rangle \right\} \end{aligned} \quad (37)$$

where we have used $t_\theta(\theta) = (\eta_{zx}^0(\theta), -\log Z_{zx}(\eta_{zx}^0(\theta)))$ in Eq. (C.4). Additionally, let $t_{zx}(z, x)$ be a multilinear polynomial in the statistics $t_z(z)$ and $t_x(x)$, and let $p(z | \theta)$ and $p(x | z, \theta)$ be a conjugate pair of exponential families, writing

$$\begin{aligned} p(z | \theta) &= \exp \left\{ \langle \eta_z^0(\theta), t_z(z) \rangle - \log Z_z(\eta_z^0(\theta)) \right\}, \\ p(x | z, \theta) &= \exp \left\{ \langle \eta_x^0(z, \theta), t_x(x) \rangle - \log Z_x(\eta_x^0(z, \theta)) \right\} \\ &= \exp \left\{ \langle t_z(z), \eta_x^0(\theta)^T (t_x(x), 1) \rangle \right\}. \end{aligned}$$

Let $p(y | x, \gamma)$ be a general family of densities (not necessarily an exponential family) and let $p(\gamma)$ be an exponential family prior on its parameters of the form

$$p(\gamma) = \exp \left\{ \langle \eta_\gamma^0, t_\gamma(\gamma) \rangle - \log Z_\gamma(\eta_\gamma^0) \right\}.$$

The corresponding variational factors are

$$\begin{aligned} q(\theta) &= \exp \left\{ \langle \eta_\theta, t_\theta(\theta) \rangle - \log Z_\theta(\eta_\theta) \right\}, & q(\gamma) &= \exp \left\{ \langle \eta_\gamma, t_\gamma(\gamma) \rangle - \log Z_\gamma(\eta_\gamma) \right\}, \\ q(z) &= \exp \left\{ \langle \eta_z, t_z(z) \rangle - \log Z_z(\eta_z) \right\}, & q(x) &= \exp \left\{ \langle \eta_x, t_x(x) \rangle - \log Z_x(\eta_x) \right\}. \end{aligned}$$

As in Section D.1, we construct the surrogate objective $\widehat{\mathcal{L}}$ to allow us to exploit exponential family and conjugacy structure. In particular, we construct $\widehat{\mathcal{L}}$ to resemble the mean field objective, namely

$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_z, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(z)q(x)} \left[\log \frac{p(\theta)p(\gamma)p(z | \theta)p(x | z, \theta)p(y | x, \gamma)}{q(\theta)q(\gamma)q(z)q(x)} \right],$$

but in $\widehat{\mathcal{L}}$ we replace the $\log p(y | x, \gamma)$ likelihood term, which may be a general family of densities without much structure, with a more tractable approximation,

$$\widehat{\mathcal{L}}(\eta_\theta, \eta_z, \eta_x, \phi) \triangleq \mathbb{E}_{q(\theta)q(z)q(x)} \left[\log \frac{p(\theta)p(z | \theta)p(x | z, \theta) \exp\{\psi(x; y, \phi)\}}{q(\theta)q(z)q(x)} \right],$$

where $\psi(x; y, \phi)$ is a function on x that resembles a conjugate likelihood for $p(x | z, \theta)$,

$$\psi(x; y, \phi) \triangleq \langle r(y; \phi), t_x(x) \rangle, \quad \phi \in \mathbb{R}^m.$$

We then define $\eta_z^*(\eta_\theta, \phi)$ and $\eta_x^*(\eta_\theta, \phi)$ to be local partial optimizers of $\widehat{\mathcal{L}}$ given fixed values of the other parameters η_θ and ϕ , and in particular they satisfy the first-order necessary optimality conditions

$$\nabla_{\eta_z} \widehat{\mathcal{L}}(\eta_\theta, \eta_z^*(\eta_\theta, \phi), \eta_x^*(\eta_\theta, \phi), \phi) = 0, \quad \nabla_{\eta_x} \widehat{\mathcal{L}}(\eta_\theta, \eta_z^*(\eta_\theta, \phi), \eta_x^*(\eta_\theta, \phi), \phi) = 0.$$

The SVAE objective is then

$$\mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \triangleq \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_z^*(\eta_\theta, \phi), \eta_x^*(\eta_\theta, \phi)). \quad (38)$$

The structure of the surrogate objective $\widehat{\mathcal{L}}$ is chosen so that it resembles the mean field variational inference objective for the conditionally conjugate model of Section C.4, and as a result we can use the same block coordinate ascent algorithm to efficiently find partial optimizers $\eta_z^*(\eta_\theta, \phi)$ and $\eta_x^*(\eta_\theta, \phi)$.

Proposition D.6 (Computing $\eta_z^*(\eta_\theta, \phi)$ and $\eta_x^*(\eta_\theta, \phi)$)

Let the densities $p(\theta, \gamma, z, x, y)$ and $q(\theta)q(\gamma)q(z)q(x)$ and the objectives \mathcal{L} , $\widehat{\mathcal{L}}$, and $\mathcal{L}_{\text{SVAE}}$ be as in Eqs. (D.4)-(D.4). The partial optimizers η_z^* and η_x^* , defined by

$$\eta_z^* \triangleq \arg \max_{\eta_z} \widehat{\mathcal{L}}(\eta_\theta, \eta_z, \eta_x, \phi), \quad \eta_x^* \triangleq \arg \max_{\eta_x} \widehat{\mathcal{L}}(\eta_\theta, \eta_z, \eta_x, \phi)$$

with the other arguments fixed, are given by

$$\eta_z^* = \mathbb{E}_{q(\theta)} \eta_z^0(\theta) + \mathbb{E}_{q(\theta)q(x)} \eta_x^0(\theta)^T(t_x(x), 1), \quad \eta_x^* = \mathbb{E}_{q(\theta)q(z)} \eta_x^0(z, \theta) + r(y; \phi), \quad (39)$$

and by alternating the expressions in Eq. (D.6) as updates we can compute $\eta_z^*(\eta_\theta, \phi)$ and $\eta_x^*(\eta_\theta, \phi)$ as local partial optimizers of $\widehat{\mathcal{L}}$.

Proof. These updates follow immediately from Lemma C.2. Note in particular that the stationary conditions $\nabla_{\eta_z} \widehat{\mathcal{L}} = 0$ and $\nabla_{\eta_x} \widehat{\mathcal{L}} = 0$ yield the each expression in Eq. (D.6), respectively. \square

The other properties developed in Propositions D.2, D.3, and D.5 also hold true for this model because it is a special case in which we have separated out the local variables, denoted x in earlier sections, into two groups, denoted z and x here, to match the exponential family structure in $p(z | \theta)$ and $p(x | z, \theta)$, and performed unconstrained optimization in each of the variational parameters. However, the expression for the natural gradient is slightly simpler for this model than the corresponding version of Proposition D.3. For completeness, we restate Proposition D.3 using the notation of this section.

Proposition D.7 (Natural gradient of the SVAE objective)

The natural gradient of the SVAE objective Eq. (D.4) with respect to the conjugate global variational parameters η_θ is

$$\begin{aligned} \tilde{\nabla}_{\eta_\theta} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) &= (\eta_\theta^0 + \mathbb{E}_{q^*(z)q^*(x)} [(t_{zx}(z, x), 1)] - \eta_\theta) \\ &\quad + (\nabla_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_z^*(\eta_\theta, \phi), \eta_x^*(\eta_\theta, \phi)), 0). \end{aligned}$$

Proof. Note that the optimality condition satisfied by η_z^* , namely

$$\nabla_{\eta_z} \widehat{\mathcal{L}}(\eta_\theta, \eta_z^*(\eta_\theta, \phi), \eta_x^*(\eta_\theta, \phi), \phi) = \nabla_{\eta_z} \mathbb{E}_{q(\theta)q(z)q(x)} \left[\log \frac{p(z | \theta)p(x | z, \theta)}{q(z)} \right] = 0,$$

also implies that it is stationary for \mathcal{L} ,

$$\nabla_{\eta_z} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_z^*(\eta_\theta, \phi), \eta_x^*(\eta_\theta, \phi)) = \nabla_{\eta_z} \mathbb{E}_{q(\theta)q(z)q(x)} \left[\log \frac{p(z | \theta)p(x | z, \theta)}{q(z)} \right] = 0,$$

and so by Proposition A.3 the term involving $\nabla_{\eta_z} \mathcal{L}$ does not appear in the chain rule expansion for the gradient $\nabla_{\eta_z} \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$. The remainder of the proof follows that of Proposition D.3. \square

E Experiment details and expanded figures

For the synthetic 1D dot video data, we trained an LDS SVAE on 80 random image sequences each of length 50, using one sequence per update, and show the model's future predictions given a prefix of a longer sequence. We used MLP image and recognition models each with one hidden layer of 50 units and a latent state space of dimension 8. The middle and bottom panels of Fig. 8 show the model's predictions and sampled latent state trajectories, respectively, with the predictions conditioned on the data up to the vertical red line. The model is able both to represent the image accurately and to make long-term predictions while modeling uncertainty.

See also videos of [training a warped mixture](#) and [training a nonlinear LDS](#).

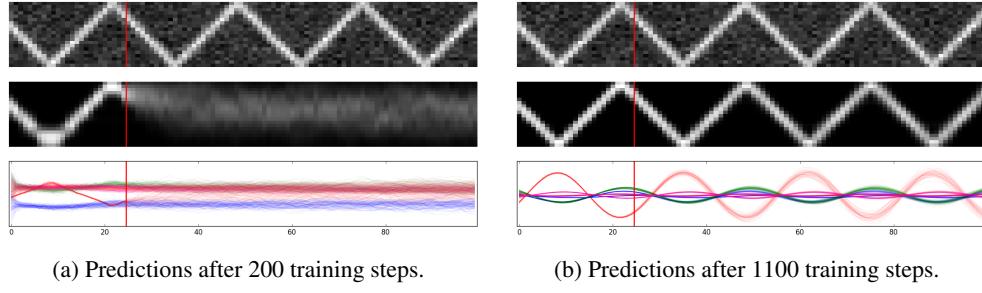


Figure 8: Predictions from an LDS SVAE fit to 1D dot image data at two stages of training. The top panel shows an example sequence with time on the horizontal axis. The middle panel shows the noiseless predictions given data up to the vertical line, while the bottom panel shows the latent states.

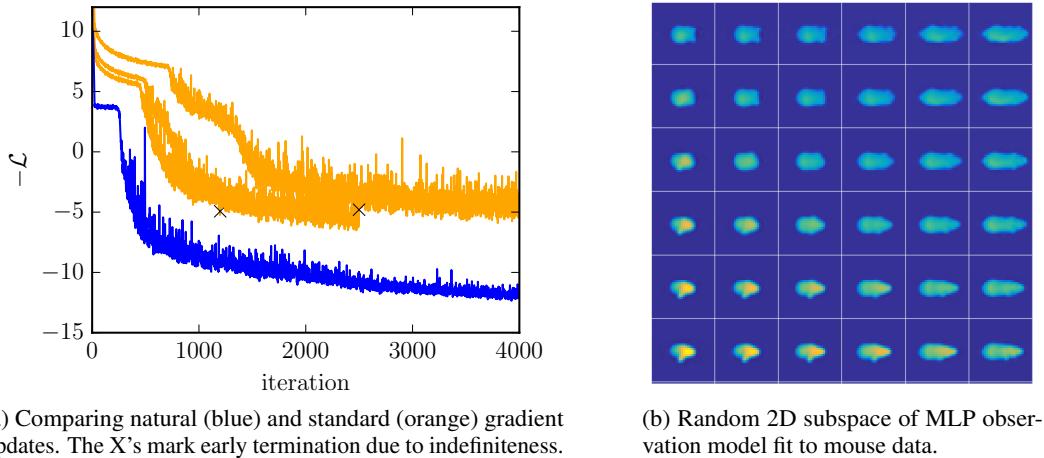


Figure 9: Panel (a) compares natural and standard gradient updates and Panel (b) shows a random 2D subspace in the image manifold coordinates learned by fitting a VAE to mouse depth video data.

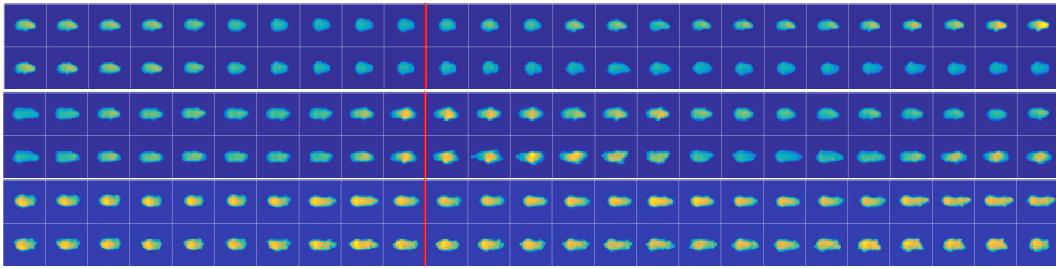
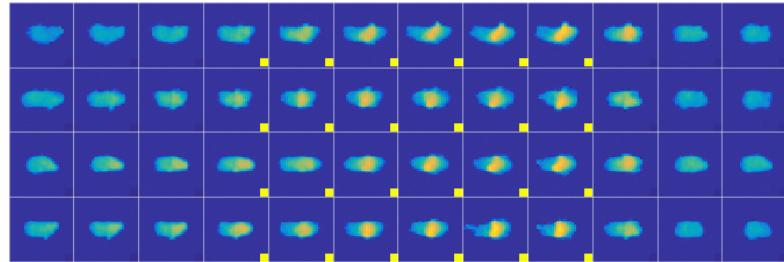
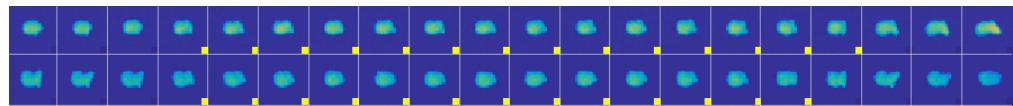


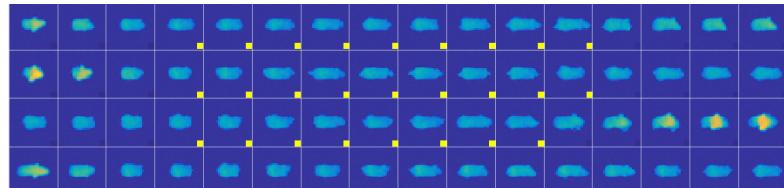
Figure 10: Examples of predictions from an LDS SVAE fit to depth video. In each panel, the top row is a sampled prediction from the LDS SVAE and the bottom row is real data. To the left of the line, the model is conditioned on the corresponding data frames and hence generates denoised versions of the same images. To the right of the line, the model is not conditioned on the data, thus illustrating the model's predictions. The frame sequences are temporally subsampled to reduce their length, showing one of every four video frames.



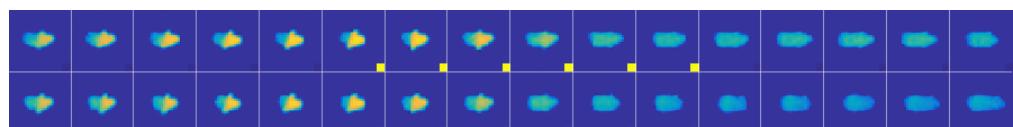
(a) Beginning a rear



(b) Grooming



(c) Extension into running



(d) Fall from rear

Figure 11: Examples of behavior states inferred from depth video. For each state, four example frame sequences are shown, including frames during which the given state was most probable according to the variational distribution on the hidden state sequence. Each frame sequence is padded on both sides, with a square in the lower-right of a frame depicting that the state was active in that frame. The frame sequences are temporally subsampled to reduce their length, showing one of every four video frames. Examples were chosen to have durations close to the median duration for that state.