Variational Autoencoder for Deep Learning of Images, Labels and Captions: Supplementary Material

Yunchen Pu[†], **Zhe Gan**[†], **Ricardo Henao**[†], **Xin Yuan**[‡], **Chunyuan Li**[†], **Andrew Stevens**[†] **and Lawrence Carin**[†]
[†]Department of Electrical and Computer Engineering, Duke University
{yp42, zg27, r.henao, cl319, ajs104, lcarin}@duke.edu
[‡]Nokia Bell Labs, Murray Hill
xyuan@bell-labs.com

1 Semi-supervised Results on ImageNet 2012

Table 1: Semi-supervised classification accuracy (%) on the validation set of ImageNet 2012.

Proportion	1%	5%	10%	20%	30%	40%
top-1						
AlexNet	0.1± 0.01	11.5 ± 0.72	19.8 ± 0.71	38.6 ± 0.31	43.23 ± 0.28	45.85 ± 0.23
GoogeLeNet	4.75± 0.58	22.13± 1.14	32.18± 0.80	42.83± 0.28	49.61± 0.11	51.90 ± 0.20
BSVM (ours)	43.98± 1.15	47.36± 0.91	48.41± 0.76	51.51± 0.28	54.14± 0.12	57.34± 0.18
Softmax (ours)	42.89	46.42	47.51	50.75	53.49	56.83
top-5						
AlexNet	0.5 ± 0.01	25.5 ± 0.92	38.60 ± 0.90	55.58 ± 0.25	63.12 ± 0.23	66.53 ± 0.22
GoogeLeNet	11.33 ± 0.96	41.33 ± 1.34	56.33 ± 0.86	68.33 ± 0.21	74.50 ± 0.12	76.94 ± 0.14
Ours	60.57 ± 1.61	62.67 ± 1.14	64.76 ± 0.90	75.67 ± 0.19	78.95 ± 0.10	80.94 ± 0.13
Softmax (ours)	59.20	61.40	63.58	74.96	78.39	80.46

Proportion	50%	60%	70%	80%	90%	100%
top-1						
AlexNet	48.25 ± 0.23	50.34 ± 0.18	52.12 ± 0.14	53.97 ± 0.14	55.62 ± 0.09	57.1
GoogeLeNet	55.09 ± 0.23	57.78 ± 0.23	61.25 ± 0.15	63.82 ± 0.17	66.18 ± 0.05	68.7
BSVM (ours)	59.73± 0.21	61.24± 0.19	61.72± 0.14	61.77± 0.13	61.79± 0.04	61.8
Softmax (ours)	59.33	60.91	61.40	61.44	61.49	61.53
top-5						
AlexNet	69.43 ± 0.18	72.18 ± 0.19	74.81 ± 0.13	77.06 ± 0.13	78.87 ± 0.09	80.2
GoogeLeNet	79.44 ± 0.17	81.70 ± 0.11	83.87 ± 0.14	84.97 ± 0.18	86.6 ± 0.09	88.9
BSVM (ours)	81.15 ± 0.13	82.53 ± 0.10	83.2 ± 0.12	83.65 ± 0.17	83.91 ± 0.08	84.3
Softmax (ours)	80.68	82.12	82.82	83.13	83.51	83.88

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

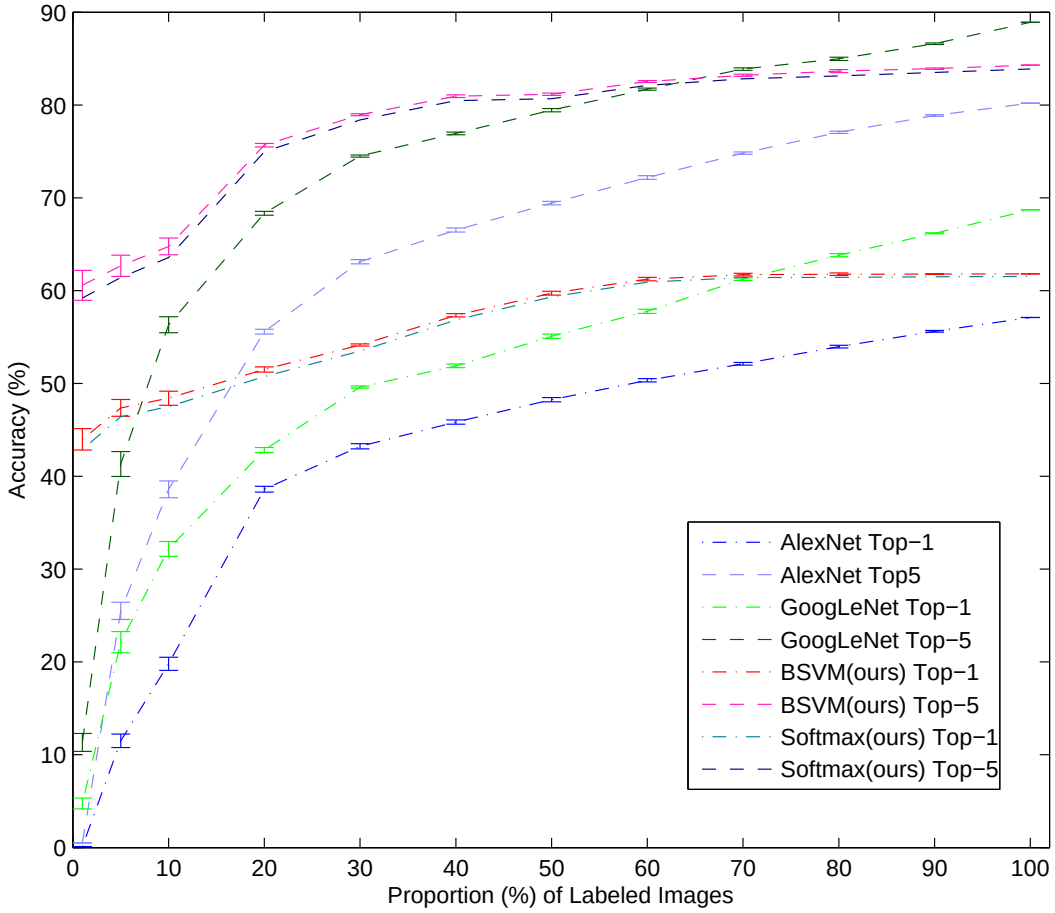

Figure 1: Semi-supervised classification accuracy on the validation set of ImageNet 2012.

Table 2: Architecture of the image models. Image Size: spatial size $\times$ color channel (one for gray and three for RGB), *e.g.*, $28^2 \times 1$. Dictionary: dictionary number $\times$ dictionary spatial size, *e.g.*, 30×8^2. Pooling: pooling/unpooling window size, *e.g.*, 3×3.

Dataset	Image Size	Model Architecture					
			Layer-1	Layer-2	Layer-3	Layer-4	Layer-5
MNIST	$28^2 \times 1$	Dictionary	30×8^2	80×6^2	-	-	-
		Pooling	3×3	-	-	-	-
CIFAR-10	$32^2 \times 3$	Dictionary	48×5^2	128×5^2	128×5^2	-	-
		Pooling	2×2	2×2	-	-	-
CIFAR-100	$32^2 \times 3$	Dictionary	48×5^2	128×5^2	128×5^2	-	-
		Pooling	2×2	2×2	-	-	-
Caltech 101	$128^2 \times 3$	Dictionary	48×7^2	84×5^2	84×5^2	-	-
		Pooling	4×4	2×2	-	-	-
Caltech 256	$128^2 \times 3$	Dictionary	48×7^2	128×5^2	128×5^2	-	-
		Pooling	4×4	2×2	-	-	-
ImageNet	$256^2 \times 3$	Dictionary	96×5^2	256×5^2	512×5^2	1024×5^2	512×5^2
		Pooling	4×4	2×2	2×2	2×2	-
Flickr8k	$256^2 \times 3$	Dictionary	48×5^2	84×5^2	128×5^2	192×5^2	128×5^2
		Pooling	4×4	2×2	2×2	2×2	-
Flickr30k	$256^2 \times 3$	Dictionary	48×5^2	84×5^2	128×5^2	384×5^2	256×5^2
		Pooling	4×4	2×2	2×2	2×2	-
MS COCO	$256^2 \times 3$	Dictionary	48×5^2	84×5^2	128×5^2	512×5^2	384×5^2
		Pooling	4×4	2×2	2×2	2×2	-

2 Model Architecture and Initialization

The architecture of the image models for each dataset in all the experiments are summarized in Table 2. For example, MNIST data is composed of gray images with spatial size 28×28 and CIFAR-10 is composed of RGB color images with spatial size 32×32. A two-layer model is used with dictionary element size 8×8 and 6×6 at the first and second layer, respectively. The pooling size is 3×3 ($p_x = p_y = 3$) and the number of dictionary elements at layers 1 and 2 are $K_1 = 30$ and $K_2 = 80$, respectively.

All the parameters for the image model are initialized at random and we do not perform layer-wise pretraining as in [1]. For the RNN training employed in image captioning, we initialize all recurrent matrices with orthogonal initialization as suggested in [2]. Non-recurrent weights are initialized from an uniform distribution in [-0.01,0.01]. All the bias terms are initialized to zero. Word vectors are initialized with the publicly available *word2vec* vectors that were trained on 100 billion words from Google News, these vectors have dimensionality 300 and were trained using a continuous bag-of-words architecture [3]. Words not present in the set of pretrained words are initialized at random. The number of hidden units in the RNNs is set to 512.

3 Details for the Variational Autoencoder

3.1 Image Captioning

Recall the variational lower bound for image captioning:

$$\mathcal{L}(\mathbf{X}, \mathbf{Y}) = \xi\big\{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})}[\log p_{\boldsymbol{\psi}}(\mathbf{Y}|\boldsymbol{s})]\big\} + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s},\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X}, \boldsymbol{s}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{s}, \boldsymbol{z}|\mathbf{X})] \quad (1)$$

The gradient of the variational lower bound w.r.t to the decoder model parameters is straightforward:

$$\nabla_{\boldsymbol{\psi}}\mathcal{L}(\mathbf{X}, \mathbf{Y}) = \xi\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})}[\nabla_{\boldsymbol{\psi}} \log p_{\boldsymbol{\psi}}(\mathbf{Y}|\boldsymbol{s})] \quad (2)$$

$$\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\mathbf{X}, \mathbf{Y}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s},\boldsymbol{z}|\mathbf{X})}[\nabla_{\boldsymbol{\alpha}} \log p_{\boldsymbol{\alpha}}(\mathbf{X}|\boldsymbol{s}, \boldsymbol{z})] \quad (3)$$

The corresponding gradient w.r.t the encoder model is

$$\begin{aligned}\nabla_{\boldsymbol{\phi}}\mathcal{L}(\mathbf{X}, \mathbf{Y}) = {} & \xi\big\{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})}[\log p_{\boldsymbol{\psi}}(\mathbf{Y}|\boldsymbol{s})] \times \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})\big\} \\ & + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s},\boldsymbol{z}|\mathbf{X})}\big\{[\log p_{\boldsymbol{\alpha}}(\mathbf{X}|\boldsymbol{s}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{s}, \boldsymbol{z}|\mathbf{X})] \times \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{s}, \boldsymbol{z}|\mathbf{X})\big\} \quad (4)\end{aligned}$$

If we use Monte Carlo integration to approximate the expectation in (4), the variance of the estimator can be very high. Since there are both real and binary latent variables in (1), we use the variance reduction techniques in [4] and [5]. The variational lower bound in (1) can be expressed as

$$
\begin{aligned}
&\mathcal{L}(\mathbf{X},\mathbf{Y}) = \qquad (5)\\
&= \xi\big\{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})}[\log p_{\boldsymbol{\psi}}(\mathbf{Y}|\boldsymbol{s})]\big\} + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s},\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X},\boldsymbol{z}|\boldsymbol{s}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{s}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})]\\
&= \xi\big\{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})}[\log p_{\boldsymbol{\psi}}(\mathbf{Y}|\boldsymbol{s})]\big\} - D_{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})||p_{\boldsymbol{\alpha}}(\boldsymbol{s})] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s},\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X},\boldsymbol{z}|\boldsymbol{s}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})]\\
&= -D_{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})||p_{\boldsymbol{\alpha}}(\boldsymbol{s})] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})}\Big\{\xi[\log p_{\boldsymbol{\psi}}(\mathbf{Y}|\boldsymbol{s})] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X},\boldsymbol{z}|\boldsymbol{s}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})]\Big\}
\end{aligned}
$$

Recall that $q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(L)}), \text{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(L)})))$ and $p(\boldsymbol{s}) = \mathcal{N}(\mathbf{0},\mathbf{I})$. Assume J is the dimension of $\boldsymbol{z}$, and μ_j and σ_j is the jth element of $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(L)})$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(L)})$, respectively. We can get the closed form of the KL term:

$$
-D_{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})||p_{\boldsymbol{\alpha}}(\boldsymbol{s})] = \frac{1}{2}\sum_{j=1}^{J}\big\{(1-(\mu_j)^2 - (\sigma_j)^2 + \log\big((\sigma_j)^2\big)\big\} \qquad (6)
$$

Using the reparameterization trick in [4]

$$
\boldsymbol{s} = f(\boldsymbol{\phi},\boldsymbol{\epsilon}) = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(L)}) + \boldsymbol{\epsilon}(\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(L)}), \quad \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}) \qquad (7)
$$

The expectation term can be expressed as

$$
\begin{aligned}
&\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})}\Big\{\xi[\log p_{\boldsymbol{\psi}}(\mathbf{Y}|\boldsymbol{s})] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X},\boldsymbol{z}|\boldsymbol{s}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})]\Big\} \qquad (8)\\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})}\Big\{\xi[\log p_{\boldsymbol{\psi}}(\mathbf{Y}|\boldsymbol{s}=f(\boldsymbol{\phi},\boldsymbol{\epsilon}))] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X},\boldsymbol{z}|\boldsymbol{s}=f(\boldsymbol{\phi},\boldsymbol{\epsilon})) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})]\Big\}
\end{aligned}
$$

Therefore, the gradient of the lower bound with respect to $\boldsymbol{\phi}$ can be expressed as

$$
\begin{aligned}
\nabla_{\boldsymbol{\phi}}\mathcal{L}(\mathbf{X},\mathbf{Y}) = &-\nabla_{\boldsymbol{\phi}} D_{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{s}|\mathbf{X})||p_{\boldsymbol{\alpha}}(\boldsymbol{s})] \qquad (9)\\
&+ \mathbb{E}_{p(\boldsymbol{\epsilon})}\Big\{\nabla_{\boldsymbol{\phi}}\xi[\log p_{\boldsymbol{\psi}}(\mathbf{Y}|\boldsymbol{s}=f(\boldsymbol{\phi},\boldsymbol{\epsilon}))] \qquad (10)\\
&+ \nabla_{\boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X},\boldsymbol{z}|\boldsymbol{s}=f(\boldsymbol{\phi},\boldsymbol{\epsilon})) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})]\Big\} \qquad (11)
\end{aligned}
$$

This expectation can approximated by Monto Carlo sampling:

$$
\begin{aligned}
&\frac{1}{N_s}\sum_{i=1}^{N_s}\Big\{\nabla_{\boldsymbol{\phi}}\xi[\log p_{\boldsymbol{\psi}}(\mathbf{Y}|\boldsymbol{s}=f(\boldsymbol{\phi},\boldsymbol{\epsilon}_i))] \qquad (12)\\
&\qquad + \nabla_{\boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X},\boldsymbol{z}|\boldsymbol{s}=f(\boldsymbol{\phi},\boldsymbol{\epsilon}_i)) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})]\Big\} \qquad (13)
\end{aligned}
$$

where $\nabla_{\boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X},\boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})]$ is the same gradient as in [5].

3.2 Image Classification

Recall that the pseudo-likelihood of a label $\ell_n \in \{1,\ldots,C\}$

$$
\begin{aligned}
\mathcal{L}(\ell_n|\boldsymbol{s}_n,\boldsymbol{\beta},\gamma) &= \prod_{\ell=1}^{C}(y_n^{(\ell)}|\boldsymbol{s}_n,\boldsymbol{\beta}_\ell,\gamma_\ell) \qquad (14)\\
&= \prod_{\ell=1}^{C}\Big\{\int_0^\infty \frac{\sqrt{\gamma_\ell}}{\sqrt{2\pi\lambda_n^{(\ell)}}}\exp\left(-\frac{(1+\lambda_n^{(\ell)} - y_n^{(\ell)}\boldsymbol{\beta}_\ell^T\boldsymbol{s}_n)^2}{2\gamma_\ell^{-1}\lambda_n^{(\ell)}}\right)d\lambda_n^{(\ell)}\Big\} \qquad (15)
\end{aligned}
$$

$\boldsymbol{\beta}$ is treated as another model parameter (part of ψ). $\lambda_n^{(\ell)}$ is treated as latent variable. We have

$$
\begin{aligned}
p(\ell_n,\boldsymbol{\lambda}_n|\boldsymbol{s}_n,\boldsymbol{\beta},\gamma) &= \prod_{\ell=1}^{C}(y_n^{(\ell)}|\boldsymbol{s}_n,\lambda_n^{(\ell)},\boldsymbol{\beta}_\ell,\gamma_\ell) \qquad (16)\\
&= \prod_{\ell=1}^{C}\Big\{\frac{\sqrt{\gamma_\ell}}{\sqrt{2\pi\lambda_n^{(\ell)}}}\exp\left(-\frac{(1+\lambda_n^{(\ell)} - y_n^{(\ell)}\boldsymbol{\beta}_\ell^T\boldsymbol{s}_n)^2}{2\gamma_\ell^{-1}\lambda_n^{(\ell)}}\right)\Big\} \qquad (17)
\end{aligned}
$$

Therefore, the variational lower bound for image classification is

$$\begin{aligned}\mathcal{L}(\mathbf{X},\mathbf{Y}) = \ & \xi\big\{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}_n,\boldsymbol{\lambda}_n|\mathbf{X}_n,\ell_n)}[\log p_{\boldsymbol{\psi}}(\boldsymbol{\lambda}_n,\ell_n|\boldsymbol{s})]\big\} \\ & + \ \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s},\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X},\boldsymbol{s},\boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{s},\boldsymbol{z}|\mathbf{X})] \end{aligned} \tag{18}$$

Since for the most part (18) is the same as for the image caption model, we only discuss the gradient of the lower bound w.r.t. $\boldsymbol{\beta}$. The first term of variational lower bound can be expressed as

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}_n,\boldsymbol{\lambda}_n|\mathbf{X}_n,\ell_n)}[\log p_{\boldsymbol{\psi}}(\boldsymbol{\lambda}_n,\ell_n|\boldsymbol{s})] = \sum_{\ell=1}^{C}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}_n,\boldsymbol{\lambda}_n^{(\ell)}|\mathbf{X}_n,y_n^{(\ell)})}[\log p_{\boldsymbol{\psi}}(\boldsymbol{\lambda}_n^{(\ell)},y_n^{(\ell)}|\boldsymbol{s}_n)] \tag{19}$$

Note that $q_{\boldsymbol{\phi}}(\boldsymbol{s}_n,\boldsymbol{\lambda}_n|\mathbf{X}_n,y_n^{(\ell)}) = q_{\boldsymbol{\phi}}(\boldsymbol{s}_n|\mathbf{X}_n)q_{\boldsymbol{\phi}}(\boldsymbol{\lambda}_n|y_n^{(\ell)})$, hence we get

$$\sum_{\ell=1}^{C}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}_n,\boldsymbol{\lambda}_n^{(\ell)}|\mathbf{X}_n,y_n^{(\ell)})}[\log p_{\boldsymbol{\psi}}(\boldsymbol{\lambda}_n^{(\ell)},y_n^{(\ell)}|\boldsymbol{s}_n)] \tag{20}$$

$$= \sum_{\ell=1}^{C}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{s}_n|\mathbf{X}_n)}\Big\{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\lambda}_n^{(\ell)}|y_n^{(\ell)})}[\log p_{\boldsymbol{\psi}}(\boldsymbol{\lambda}_n^{(\ell)},y_n^{(\ell)}|\boldsymbol{s}_n)]\Big\} \tag{21}$$

Since

$$\log p_{\boldsymbol{\psi}}(\boldsymbol{\lambda}_n^{(\ell)},y_n^{(\ell)}|\boldsymbol{s}_n) = -\frac{(1+\lambda_n^{(\ell)}-y_n^{(\ell)}\boldsymbol{\beta}_\ell^T\boldsymbol{s}_n)^2}{2\gamma_\ell^{-1}\lambda_n^{(\ell)}} + c(\boldsymbol{\lambda}_n^{(\ell)},y_n^{(\ell)},\gamma_\ell) \tag{22}$$

where $c(\boldsymbol{\lambda}_n^{(\ell)},y_n^{(\ell)},\gamma_\ell)$ are independent of β_ℓ, we can find that the relevant portion of Equation (22) is a linear function of $(\boldsymbol{\lambda}_n^{(\ell)})^{-1}$. It means the expectation term $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\lambda}_n^{(\ell)}|y_n^{(\ell)})}[\log p_{\boldsymbol{\psi}}(\boldsymbol{\lambda}_n^{(\ell)},y_n^{(\ell)}|\boldsymbol{s}_n)]$ in Equation (21) can be obtained by simply replacing $(\boldsymbol{\lambda}_n^{(\ell)})^{-1}$ with its conditional expectation. From [6], we have

$$q_{\boldsymbol{\phi}}((\boldsymbol{\lambda}_n^{(\ell)})^{-1}|y_n^{(\ell)}) = \mathcal{IG}(|1-\boldsymbol{y}_n^\ell\boldsymbol{s}_n^\top\boldsymbol{\beta}^{(\ell)}|^{-1},1) \tag{23}$$

$$\mathbb{E}((\boldsymbol{\lambda}_n^{(\ell)})^{-1}) = |1-\boldsymbol{y}_n^\ell\boldsymbol{s}_n^\top\boldsymbol{\beta}^{(\ell)}|^{-1} \tag{24}$$

Thus, using the same reparameterization trick in (7), we can get the gradient w.r.t. $\boldsymbol{\beta}$.

4 Multilayer Perceptrons

$\boldsymbol{\mu}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(n,2)})$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(n,2)})$ are constituted by "stacking" the K_2 spatially aligned $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(n,k_2,2)})$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(n,k_2,2)})$, respectively, which are defined as (bias are omitted in the main paper)

$$\boldsymbol{\mu}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(n,k_2,2)}) = \mathbf{W}_{\boldsymbol{\mu}}^{(k_2)}\boldsymbol{h}^{(k_2)} + \boldsymbol{b}_{\boldsymbol{\mu}}^{(k_2)} \tag{25}$$

$$\log\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\tilde{\mathbf{C}}^{(n,k_2,2)}) = \mathbf{W}_{\boldsymbol{\phi}}^{(k_2)}\boldsymbol{h}^{(k_2)} + \boldsymbol{b}_{\boldsymbol{\phi}}^{(k_2)} \tag{26}$$

$$\boldsymbol{h}^{(k_2)} = \tanh\Big(\mathbf{W}^{(k_2)}\mathsf{vec}(\tilde{\mathbf{C}}^{(n,k_2,2)}) + \boldsymbol{b}^{(k_2)}\Big) \tag{27}$$

where $k_2 = 1,\ldots,K_2$.

References

[1] Y. Pu, X. Yuan, A. Stevens, C. Li, and L. Carin. A deep generative deconvolutional image model. In *AISTATS*, 2016.

[2] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *ICLR*, 2014.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[4] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

[5] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. In *ICML*, 2014.

[6] N. G. Polson and S. L. Scott. Data augmentation for support vector machines. *Bayes. Anal.*, 2011.