Variational Adaptive-Newton Method

Mohammad Emtiyaz Khan **Wu Lin** **Voot Tangkaratt** **Zuozhu Liu*** **Didrik Nielsen**
AIP, RIKEN, Tokyo

Abstract

We present a black-box learning method called the Variational Adaptive-Newton (VAN) method[2]. The method is especially suitable for explorative-learning tasks such as active learning and reinforcement learning. Just like Bayesian methods, it estimates a distribution that can be used for exploration, but its computational complexity is similar to that of the continuous optimization methods. VAN is a second-order method that unifies existing methods in distinct fields of continuous optimization, variational inference, and evolution strategies. Our results show that VAN performs well on a wide-variety of learning tasks but takes fewer computations than existing approaches. To summarize, this works presents a method has the potential not only to be a general-purpose learning tool, but also to be used to solve difficult problems such as deep reinforcement learning and unsupervised learning.

1 Introduction

Humans learn by sequentially exploring the world. During our lifetimes, we use our past experiences to acquire new ones and then use them to learn even more. How can we design methods that can mimic this "explorative" learning? This is an open question in artificial intelligence and machine learning.

One such approach is based on the Bayesian posterior distribution which can be used to summarize past data examples and to choose new ones (Perfors et al., 2011). Bayesian methods have been applied to a wide variety of explorative-learning tasks, e.g., for active learning and Bayesian optimization to select informative data examples (Houlsby et al., 2011; Gal et al., 2017; Brochu et al., 2010; Fishel and Loeb, 2012), and for reinforcement learning to learn through interactions (Wyatt, 1998; Strens, 2000). However, estimating the posterior distribution is computationally demanding, which makes its application to large-scale problems challenging.

On the other hand, non-Bayesian methods, such as those based on continuous optimization, are computationally much cheaper. This is because they only compute a point estimate of the model parameters rather than their distributions. However, since there is no distribution associated with the parameters, these methods cannot directly exploit the mechanisms of Bayesian exploration. This raises the following question: how can we design explorative-learning methods that compute a distribution just like Bayesian methods, but cost similar to optimization methods?

In this paper, we propose such a method. Our method is a general purpose black-box optimization method, but is especially suitable for exploration-based learning. Just like Newton's method, our method is a second-order method, but it also unifies existing methods in distinct fields of continuous optimization, variational inference, and evolution strategies. Due to this connection, we call our method the Variational Adaptive-Newton (VAN) method. We show results for supervised and unsupervised learning, as well as for active learning and reinforcement learning. Our results suggest that VAN can be used as a general purpose learning method and has the potential to improve existing approaches for a variety of difficult learning problems.

*ZL is from Singapore University of Technology and Design, Singapore. Work was done in internship at AIP.

[2]A longer version of this paper is available at `https://arxiv.org/abs/1711.05560`.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

2 Variational Optimization with Gaussian distributions

We consider learning tasks that can be formulated as a function minimization problem,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, f(\boldsymbol{\theta}), \qquad \text{where } \boldsymbol{\theta} \in \mathbb{R}^D. \tag{1}$$

A wide variety of problems in supervised, unsupervised, and reinforcement learning fall under this category. However, instead of directly solving problem (1), we take a different approach that follows the Variational Optimization (VO) framework (Staines and Barber, 2012) where we minimize,

$$\min_{\boldsymbol{\eta}} \; \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\eta})}\left[f(\boldsymbol{\theta})\right] := \mathcal{L}(\boldsymbol{\eta}), \tag{2}$$

where q is a probability distribution with parameters $\boldsymbol{\eta}$. This approach can lead us to the minimum $\boldsymbol{\theta}^*$ because $\mathcal{L}(\boldsymbol{\eta})$ is an upper bound on the minimum value of f at $\boldsymbol{\theta}^*$, i.e., $\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \leq \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\eta})}\left[f(\boldsymbol{\theta})\right]$. Therefore minimizing $\mathcal{L}(\boldsymbol{\eta})$ minimizes $f(\boldsymbol{\theta})$, and when the distribution q puts all its mass on $\boldsymbol{\theta}^*$, we recover the minimum $\boldsymbol{\theta}^*$. This type of function minimization is commonly used in many areas of stochastic search such as evolution strategies (Hansen and Ostermeier, 2001; Wierstra et al., 2008). In our problem context, this formulation is advantageous because it enables learning via exploration, where exploration is facilitated with the use of the distribution $q(\boldsymbol{\theta}|\boldsymbol{\eta})$.

In this paper, we will use a Gaussian distribution for exploration, i.e., we set $q(\boldsymbol{\theta}|\boldsymbol{\eta}) := \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\eta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$. The problem (2) can then be rewritten as follows,

$$\min_{\mu,\Sigma} \mathbb{E}_{\mathcal{N}(\theta|\mu,\Sigma)}\left[f(\boldsymbol{\theta})\right] := \mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{3}$$

A straightforward approach to minimize $\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is to use SGD to optimize $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as shown below,

$$\text{V-SGD} \;:\quad \boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \rho_t \left[\widehat{\nabla}_\mu \mathcal{L}_t\right], \qquad \boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t - \rho_t \left[\widehat{\nabla}_\Sigma \mathcal{L}_t\right], \tag{4}$$

where $\rho_t > 0$ is a step size at iteration t, $\widehat{\nabla}$ denotes an unbiased stochastic-gradient estimate, and $\mathcal{L}_t = \mathcal{L}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$. We refer to this approach as Variational SGD or simply V-SGD to differentiate it from the standard SGD to optimize $f(\boldsymbol{\theta})$ in the $\boldsymbol{\theta}$ space.

This approach is simple and can be powerful when used with adaptive-gradient methods to adaptively change the step-size, e.g., AdaGrad and RMSprop. However, as pointed by Wierstra et al. (2008), it has issues, especially when REINFORCE (Williams, 1992) is used to estimate the gradients of $f(\boldsymbol{\theta})$. Wierstra et al. (2008) argue that the V-SGD update becomes increasingly unstable when the covariance is small, while becoming very small when the covariance is large. To fix these problems, Wierstra et al. (2008) proposed a natural-gradient method. Our method is also a natural-gradient method, but, as we show in the next section, its updates are much simpler and they lead to a second-order method which is similar to Newton's method.

3 Variational Adaptive-Newton Method

We propose a new method to solve (3) by using a mirror-descent algorithm. We show that our algorithm is a second-order method which solves the original problem (1), even though it is designed to solve (3). Due to its similarity to Newton's method, we refer to our method as the Variational Adaptive-Newton (VAN) method. Figure (b) shows an illustrative example of our method to optimize a one-dimensional non-convex function.

VAN can be obtained by making two small changes to the V-SGD objective. Firstly, note that the V-SGD update in (4) is a solution of the following optimization problem:

$$\boldsymbol{\eta}_{t+1} = \underset{\boldsymbol{\eta}}{\operatorname{argmin}} \; \left\langle \boldsymbol{\eta}, \widehat{\nabla}_\eta \mathcal{L}_t \right\rangle + \frac{1}{2\rho_t}\|\boldsymbol{\eta} - \boldsymbol{\eta}_t\|^2 \quad \Rightarrow \quad \boldsymbol{\eta}_t - \rho_t \widehat{\nabla}_\eta \mathcal{L}_t. \tag{5}$$

A simple interpretation of this optimization problem is that, in V-SGD, we choose the next point $\boldsymbol{\eta}$ along the gradient but contained within a scaled ℓ_2-ball centered at the current point $\boldsymbol{\eta}_t$. This interpretation enables us to slightly modify V-SGD to obtain VAN.

Our VAN method is based on two modifications to (5). The first modification is to replace the Euclidean distance $\|\cdot\|^2$ by a *Bregman* divergence which results in the *mirror-descent* method. Note

that for exponential-family distributions, the Bregman divergence corresponds to the *Kullback-Leibler* (KL) divergence (Raskutti and Mukherjee, 2015). Using the KL divergence as distance measure has the benefit of resulting in natural gradient updates which are efficient when optimizing the parameter of a probability distribution (Amari, 1998). Our second modification is that we optimize the VO objective with respect to the mean parameterization of the Gaussian distribution $\mathbf{m} := \{\boldsymbol{\mu}, \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma}\}$ instead of the parameter $\boldsymbol{\eta} := \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$. The two modifications give us the following problem:

$$\mathbf{m}_{t+1} = \operatorname*{argmin}_{m} \left\langle \mathbf{m}, \widehat{\nabla}_m \mathcal{L}_t \right\rangle + \frac{1}{\beta_t} \mathbb{D}_{KL}[q \,\|\, q_t], \tag{6}$$

where $q := q(\boldsymbol{\theta}|\mathbf{m})$, $q_t := q(\boldsymbol{\theta}|\mathbf{m}_t)$, and $\mathbb{D}_{KL}[q \,\|\, q_t] = \mathbb{E}_q \log(q/q_t)$ denotes the KL divergence. The convergence of this procedure is guaranteed under mild conditions (Ghadimi et al., 2014).

The solution to this optimization problem is given by (proof given in Appendix A),

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \beta_t \; \boldsymbol{\Sigma}_{t+1} \left[\widehat{\nabla}_\mu \mathcal{L}_t \right], \qquad \boldsymbol{\Sigma}_{t+1}^{-1} = \boldsymbol{\Sigma}_t^{-1} + \; 2\beta_t \; \left[\widehat{\nabla}_\Sigma \mathcal{L}_t \right]. \tag{7}$$

The VAN update above corresponds to those of a second-order method. To see this connection, we rewrite the gradients in the update by using the following identities (Opper and Archambeau, 2009):

$$\nabla_\mu \mathcal{L}_t = \nabla_\mu \mathbb{E}_q \left[f(\boldsymbol{\theta}) \right] = \mathbb{E}_q \left[\nabla_\theta f(\boldsymbol{\theta}) \right], \qquad \nabla_\Sigma \mathbb{E}_q \left[f(\boldsymbol{\theta}) \right] = \tfrac{1}{2} \mathbb{E}_q \left[\nabla^2_{\theta\theta} f(\boldsymbol{\theta}) \right]. \tag{8}$$

By substituting these in (7) and expressing the update in terms of the precision matrix $\mathbf{P}_t := \boldsymbol{\Sigma}_t^{-1}$, we get the following updates:

$$\text{VAN:} \quad \boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \beta_t \; \mathbf{P}_{t+1}^{-1} \mathbb{E}_{q_t} \left[\nabla_\theta f(\boldsymbol{\theta}) \right], \qquad \mathbf{P}_{t+1} = \mathbf{P}_t + \; \beta_t \; \mathbb{E}_{q_t} \left[\nabla^2_{\theta\theta} f(\boldsymbol{\theta}) \right], \tag{9}$$

where $q_t := \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$. Compare this to the update of Newton's method,

$$\text{Newton's Method:} \quad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \rho_t \left[\nabla^2_{\theta\theta} f(\boldsymbol{\theta}_t) \right]^{-1} \left[\nabla_\theta f(\boldsymbol{\theta}_t) \right]. \tag{10}$$

While Newton's method scales the gradient by the inverse Hessian to obtain a step direction, VAN scales the *averaged gradients* by the precision matrix $\mathbf{P}_t$ which contains a weighted sum of the past *averaged Hessians*. Both the averaged-gradients and averaged-Hessians are the average of gradients and Hessians computed at locations sampled from q_t, respectively. The VAN update therefore uses the second-order information but, unlike Newton's method, this information is averaged over samples from the explorative distribution q and summed over past iterations.

4 Variants of VAN and Their Connections to Existing Methods

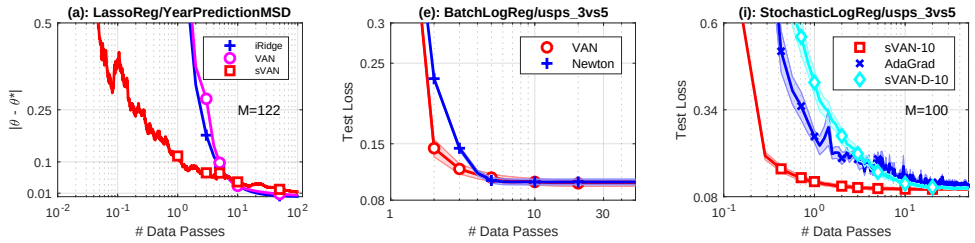
A Large-Scale Variant: For problems with a large number of parameters, computing the full Hessian matrix might be infeasible. By using a mean-field approximation $q(\boldsymbol{\theta}|\boldsymbol{\eta}) = \prod_{d=1}^{D} \mathcal{N}(\theta_d|\mu_d, \sigma_d^2)$ however, we obtain a method that use a diagonal approximation of Hessian instead of the full Hessian matrix in its updates. Denoting the precision parameters by $s_d = 1/\sigma_d^2$, and a vector containing them by $\mathbf{s}$, the following diagonal version of VAN, which we call VAN-D, can be written as follows:

$$\text{VAN-D: } \boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \beta_t \; \text{diag}(\mathbf{s}_{t+1})^{-1} \mathbb{E}_{q_t} \left[\nabla_\theta f(\boldsymbol{\theta}) \right], \qquad \mathbf{s}_{t+1} = \mathbf{s}_t + \beta_t \mathbb{E}_{q_t} \left[\mathbf{h}(\boldsymbol{\theta}) \right], \tag{11}$$

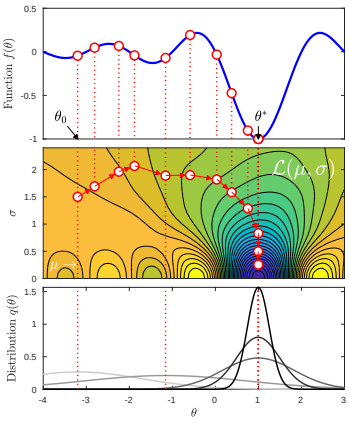
where $\text{diag}(\mathbf{s})$ is a diagonal matrix containing the vector $\mathbf{s}$ as its diagonal and $\mathbf{h}(\boldsymbol{\theta})$ is the diagonal of the Hessian $\nabla^2_{\theta\theta} f(\boldsymbol{\theta})$. The diagonal Hessian can be computed using the reparameterization trick.

Connections to AdaGrad: Instead of using the Hessian, we can use a Gauss-Newton variant of VAN-D where we replace the Hessian $\nabla^2_{\theta_i\theta_i} f(\boldsymbol{\theta})$ by the square of the gradient $\left[\nabla_{\theta_i} f(\boldsymbol{\theta})\right]^2$. The resulting updates are very similar to the updates of the adaptive-gradient method AdaGrad (Duchi et al., 2011). The main difference is that our method uses the averaged-gradients instead of the gradient at one point. VAN can also be seen as a generalization of an adaptive method called AROW (Crammer et al., 2009). AROW uses a mirror descent algorithm similar to ours, but has been applied only to SVMs.

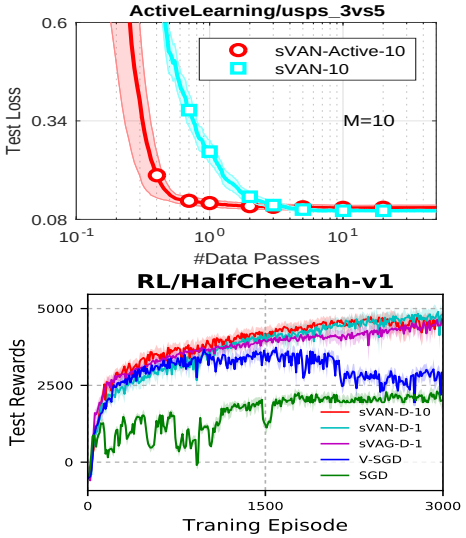
Connections to Variational Inference (VI): VI is a special type of VO problem where function $f(\boldsymbol{\theta}) := -\log[p(\mathbf{y}, \boldsymbol{\theta})/q(\boldsymbol{\theta}|\boldsymbol{\eta})]$ with $p(\mathbf{y}, \boldsymbol{\theta})$ being the joint distribution and q being the variational approximation to the posterior distribution. Our approach is an extension of a recent approach for VI by **?** called the conjugate-computation variational inference (CVI). VAN is a generalization of CVI



(a) The left figure shows improvements for Lasso regression when compared to the Iterative-Ridge method on the YearPredictionMSD dataset ($N = 515K$ and $D = 90$). Here, stochastic-VAN (sVAN) with a minibatch size of 122 converges faster than the batch methods. The middle figure shows comparison for Batch-VAN to Newton's method for USPS dataset using logistic regression ($N = 1540$ and $D = 256$), where VAN converges at the same rate as Newton's method. The right figure shows the same comparison for stochastic-VAN and its diagonal version (sVAN-D) with mini-batch size of 100 where we observe a convergence similar to AdaGrad. Both of our methods use 10 samples to compute averaged-gradients and Hessian.



(b) The top figure shows the function $f(\theta) = \text{sinc}(\theta)$ with a blue curve (global minima at 1). The second plot shows the VO objective $\mathcal{L}(\mu, \sigma) = \mathbb{E}_q[f(\theta)]$ for a Gaussian $q = \mathcal{N}(\theta|\mu, \sigma^2)$. The red points and arrows show the iterations of our VAN method initialized at $\mu = -3.2$ and $\sigma = 1.5$. The bottom plot shows the progression of q where darker curves indicate higher iterations.



(c) The top figure shows results for active learning on logistic regression. sVAN-Active-10 uses a maximum-entropy acquisition function to select a mini-batch size of 10 and achieve a good error much quickly when compared to sVAN-10 with no active learning. The bottom figure shows results for parameter-based exploration for reinforcement learning in deep deterministic-policy gradient method (Silver et al., 2014). sVAN-D variants achieve better reward than AROW, V-SGD, and SGD.

to a general function $f(\boldsymbol{\theta})$. A direct consequence of our theoretical result from the previous section is that CVI, just like VAN, is also a second-order method.

VAN as a natural-gradient method: VAN performs natural-gradient updates in the space of the Gaussian distributions. This can be shown by using a recent result by Raskutti and Mukherjee (2015). Our approach however is much simpler than existing methods such as Natural Evolution Strategies (NES) (Wierstra et al., 2008) which applies natural-gradient descent directly in the space of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Unfortunately, this results in an infeasible algorithm since the Fisher information matrix has $O(D^4)$ parameters. NES requires sophisticated reparameterization to solve this issue. An advantage of VAN over NES is that it naturally has $O(D^2)$ parameters and its updates are therefore much simpler.

Results: Our experimental results shown in Fig. (a) and (c) reveal that VAN and its variants perform well for supervised learning on Lasso and Logistic regression, active learning for logistic regression, and a reinforcement-learning task for parameter based exploration. Figure (b) shows an example where VAN can avoid local minima through exploration. These results show the generality of our method and also that it can be used to solve difficult problems such deep reinforcement learning.

References

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276.

Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Crammer, K., Kulesza, A., and Dredze, M. (2009). Adaptive regularization of weight vectors. In *Advances in neural information processing systems*, pages 414–422.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Fishel, J. A. and Loeb, G. E. (2012). Bayesian exploration for intelligent identification of textures. *Frontiers in neurorobotics*, 6.

Gal, Y., Islam, R., and Ghahramani, Z. (2017). Deep Bayesian Active Learning with Image Data. *ArXiv e-prints*.

Ghadimi, S., Lan, G., and Zhang, H. (2014). Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, pages 1–39.

Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian Active Learning for Classification and Preference Learning. *ArXiv e-prints*.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Opper, M. and Archambeau, C. (2009). The Variational Gaussian Approximation Revisited. *Neural Computation*, 21(3):786–792.

Perfors, A., Tenenbaum, J. B., Griffiths, T. L., and Xu, F. (2011). A tutorial introduction to Bayesian models of cognitive development. *Cognition*, 120(3):302–321.

Raskutti, G. and Mukherjee, S. (2015). The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. A. (2014). Deterministic policy gradient algorithms. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 387–395.

Staines, J. and Barber, D. (2012). Variational Optimization. *ArXiv e-prints*.

Strens, M. (2000). A Bayesian framework for reinforcement learning. In *In Proceedings of the Seventeenth International Conference on Machine Learning*, pages 943–950. ICML.

Wierstra, D., Schaul, T., Peters, J., and Schmidhuber, J. (2008). Natural evolution strategies. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 3381–3387. IEEE.

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

Wyatt, J. (1998). Exploration and inference in learning from reinforcement.

A Derivation of VAN

Denote the mean parameters of $q_t(\boldsymbol{\theta})$ by $\mathbf{m}_t$ which is equal to the expected value of the sufficient statistics $\boldsymbol{\phi}(\boldsymbol{\theta})$, i.e., $\mathbf{m}_t := \mathbb{E}_{q_t}[\boldsymbol{\phi}(\boldsymbol{\theta})]$. The mirror descent update at iteration t is given by the solution to

$$\mathbf{m}_{t+1} = \underset{\boldsymbol{m}}{\operatorname{argmin}} \left\langle \mathbf{m}, \widehat{\nabla}_m \mathcal{L}_t \right\rangle + \frac{1}{\beta_t} \mathbb{D}_{KL}[q \,\|\, q_t] \tag{12}$$

$$= \underset{\boldsymbol{m}}{\operatorname{argmin}}\, \mathbb{E}_q \left[\left\langle \boldsymbol{\phi}(\boldsymbol{\theta}), \widehat{\nabla}_m \mathcal{L}_t \right\rangle + \log \left((q/q_t)^{1/\beta_t} \right) \right] \tag{13}$$

$$= \underset{\boldsymbol{m}}{\operatorname{argmin}}\, \mathbb{E}_q \left[\log \frac{\exp \left\langle \boldsymbol{\phi}(\boldsymbol{\theta}), \widehat{\nabla}_m \mathcal{L}_t \right\rangle q^{1/\beta_t}}{q_t^{1/\beta_t}} \right] \tag{14}$$

$$= \underset{\boldsymbol{m}}{\operatorname{argmin}}\, \mathbb{E}_q \left[\log \left(\frac{q^{1/\beta_t}}{q_t^{1/\beta_t} \exp \left\langle \boldsymbol{\phi}(\boldsymbol{\theta}), -\widehat{\nabla}_m \mathcal{L}_t \right\rangle} \right) \right] \tag{15}$$

$$= \underset{\boldsymbol{m}}{\operatorname{argmin}} \frac{1}{\beta_t}\, \mathbb{E}_q \left[\log \left(\frac{q}{q_t \exp \left\langle \boldsymbol{\phi}(\boldsymbol{\theta}), -\beta_t \widehat{\nabla}_m \mathcal{L}_t \right\rangle} \right) \right] \tag{16}$$

$$= \underset{\boldsymbol{m}}{\operatorname{argmin}} \frac{1}{\beta_t}\, \mathbb{D}_{KL} \left[q \| q_t \exp \left\langle \boldsymbol{\phi}(\boldsymbol{\theta}), -\beta_t \widehat{\nabla}_m \mathcal{L}_t \right\rangle / \mathcal{Z}_t \right]. \tag{17}$$

where $\mathcal{Z}$ is the normalizing constant of the distribution in the denominator which is a function of the gradient and step size.

Minimizing this KL divergence gives the update

$$q_{t+1}(\boldsymbol{\theta}) \propto q_t(\boldsymbol{\theta}) \exp \left\langle \boldsymbol{\phi}(\boldsymbol{\theta}), -\beta_t \widehat{\nabla}_m \mathcal{L}_t \right\rangle. \tag{18}$$

By rewriting this, we see that we get an update in the natural parameters $\boldsymbol{\lambda}_t$ of $q_t(\boldsymbol{\theta})$, i.e.

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t - \beta_t \widehat{\nabla}_m \mathcal{L}_t. \tag{19}$$

Recalling that the mean parameters of a Gaussian $q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are $\mathbf{m}^{(1)} = \boldsymbol{\mu}$ and $\mathbf{M}^{(2)} = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T$ and using the chain rule, we can express the gradient $\widehat{\nabla}_m \mathcal{L}_t$ in terms of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$,

$$\widehat{\nabla}_{m^{(1)}} \mathcal{L} = \widehat{\nabla}_\mu \mathcal{L} - 2 \left[\widehat{\nabla}_\Sigma \mathcal{L} \right] \boldsymbol{\mu} \tag{20}$$

$$\widehat{\nabla}_{M^{(2)}} \mathcal{L} = \widehat{\nabla}_\Sigma \mathcal{L}. \tag{21}$$

Finally, recalling that the natural parameters of a Gaussian $q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are $\boldsymbol{\lambda}^{(1)} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and $\boldsymbol{\lambda}^{(2)} = -\frac{1}{2}\boldsymbol{\Sigma}^{-1}$, we can rewrite the VAN updates in terms of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$,

$$\boldsymbol{\Sigma}_{t+1}^{-1} = \boldsymbol{\Sigma}_t^{-1} + 2\beta_t \left[\widehat{\nabla}_\Sigma \mathcal{L}_t \right] \tag{22}$$

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\Sigma}_{t+1} \left[\boldsymbol{\Sigma}_t^{-1} \boldsymbol{\mu}_t - \beta_t \left(\widehat{\nabla}_\mu \mathcal{L}_t - 2 \left[\widehat{\nabla}_\Sigma \mathcal{L}_t \right] \boldsymbol{\mu}_t \right) \right] \tag{23}$$

$$= \boldsymbol{\Sigma}_{t+1} \left(\boldsymbol{\Sigma}_t^{-1} + 2\beta_t \left[\widehat{\nabla}_\Sigma \mathcal{L}_t \right] \right) \boldsymbol{\mu}_t - \beta_t \boldsymbol{\Sigma}_{t+1} \left[\widehat{\nabla}_\mu \mathcal{L}_t \right] \tag{24}$$

$$= \boldsymbol{\mu}_t - \beta_t \boldsymbol{\Sigma}_{t+1} \left[\widehat{\nabla}_\mu \mathcal{L}_t \right]. \tag{25}$$