

Projected Stein Variational Newton: A Fast and Scalable Bayesian Inference Method in High Dimensions

Peng Chen¹ Keyi Wu¹ Joshua Chen¹ Thomas O’Leary-Roseberry¹ Omar Ghattas^{1,2}

Abstract

We propose a fast and scalable variational method for Bayesian inference in high-dimensional parameter space, which we call projected Stein variational Newton (pSVN) method. We exploit the intrinsic low-dimensional geometric structure of the posterior distribution in the high-dimensional parameter space via its Hessian (of the log posterior) operator and perform a parallel update of the parameter samples projected into a low-dimensional subspace by an SVN method. The subspace is adaptively constructed using the eigenvectors of the averaged Hessian at the current samples. We demonstrate fast convergence of the proposed method and its scalability with respect to the number of parameters, samples, and processor cores.

1. Introduction

Bayesian inference provides an optimal probability formulation for learning complex models from observational or experimental data under uncertainty by updating the model parameter from its prior distribution to a posterior distribution (Stuart, 2010). In Bayesian inference we typically face the task of drawing samples from the posterior probability distribution to compute various statistics of some given quantities of interest. However, this is often prohibitive when the posterior distribution is high-dimensional; many conventional methods for Bayesian inference suffer from the curse of dimensionality, i.e., computational complexity grows exponentially or convergence deteriorates with increasing parameter dimensions.

To address this curse-of-dimensionality, several efficient and scalable methods have been developed that exploit the intrinsic properties of the posterior distribution, such as its smoothness, sparsity, and intrinsic low-dimensionality.

¹The Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712. ² Department of Mechanical Engineering, and Department of Geological Sciences, The University of Texas at Austin, Austin, TX 78712. Correspondence to: Peng Chen <peng@ices.utexas.edu>.

Markov chain Monte Carlo (MCMC) methods exploiting geometry of the log-likelihood function have been developed (Girolami & Calderhead, 2011; Martin et al., 2012; Petra et al., 2014; Cui et al., 2016; Beskos et al., 2017), providing more effective sampling than the black-box MCMC. For example, the DILI MCMC method (Cui et al., 2016) uses the low rank structure of the Hessian of the negative log likelihood in conjunction with operator-weighted proposals that are well-defined on function space to yield a sampler whose performance is dimension-independent and effective at capturing information provided by the data. However, despite these enhancements, MCMC methods remain prohibitive for problems with expensive-to-evaluate likelihoods (i.e. involving complex models) and in high parameter dimensions. Deterministic sparse quadratures were developed in (Schwab & Stuart, 2012; Schillings & Schwab, 2013; Chen & Schwab, 2015) and shown to converge rapidly with dimension-independent rates for smooth and sparse problems. However, the fast convergence is lost when the posterior distribution has significant local variations, despite the enhancements with Hessian-based transformation (Schillings & Schwab, 2016; Chen et al., 2017).

Variational inference methods reformulate the sampling problem as an optimization problem that approximates the posterior by minimizing its Kullback–Leibler divergence with a transformed prior (Marzouk et al., 2016; Liu & Wang, 2016; Blei et al., 2017), which can be potentially much faster than MCMC. In particular, Stein variational methods, which seek a composition of a sequence of simple transport maps represented by kernel functions using gradient decent (SVGD) (Liu & Wang, 2016; Chen et al., 2018; Liu & Zhu, 2018) and especially Newton (SVN) (Detommaso et al., 2018) optimization methods, are shown to achieve fast convergence in relatively low dimensions. However, these variational optimization methods can again become prohibitive in high dimensions. This can be partially addressed by a localized SVGD on Markov blankets using a sparse structure of the distribution (Wang et al., 2018).

Contributions: In this work, we develop a projected Stein variational Newton method (pSVN) to tackle the challenge of high-dimensional Bayesian inference by exploiting the intrinsic low-dimensional geometric structure of the posterior

distribution (where it departs from the prior), as characterized by the dominant spectrum of the prior-preconditioned Hessian of negative log likelihood. This low-rank structure or fast decay of eigenvalues of the preconditioned Hessian has been proven for some inference problems and commonly observed in many others with complex models (Bui-Thanh & Ghattas, 2012; Bui-Thanh et al., 2013; Spantini et al., 2015; Isaac et al., 2015; Cui et al., 2016; Chen et al., 2017; 2019; Bashir et al., 2008). By projecting the parameter into this data-informed low-dimensional subspace and applying the projected SVN in this subspace, we can effectively mitigate the curse of dimensionality. We demonstrate fast convergence of pSVN that is independent of the number of parameter dimensions and samples. We present a scalable parallel implementation of pSVN that yields rapid convergence, minimal communication, and low memory footprint, thanks to this low-dimensional projection.

Below, we present background on Bayesian inference and Stein variational methods in Section 2, develop the projected Stein variational Newton method in Section 3, and provide numerical experiments in Section 4.

2. Background

We present a general formulation of Bayesian inference problems and Stein variational methods in this section.

2.1. Bayesian inference

We consider a random parameter $x \in \mathbb{R}^d$, $d \in \mathbb{N}$, with a prior distribution μ_0 , and noisy observational data y of a parameter-to-observable map $f : \mathbb{R}^d \rightarrow \mathbb{R}^s$, $s \in \mathbb{N}$, i.e.,

$$y = f(x) + \xi, \quad (1)$$

where we consider a Gaussian noise $\xi \sim \mathcal{N}(0, \Gamma_{\text{noise}})$ with symmetric, and positive definite covariance $\Gamma_{\text{noise}} \in \mathbb{R}^{s \times s}$. The posterior distribution μ_y of the parameter x conditioned on the data y is given by Bayes' rule

$$\frac{d\mu_y}{d\mu_0}(x) = \frac{1}{Z} \exp(-\eta_y(x)), \quad (2)$$

where $\eta_y : \mathbb{R}^d \rightarrow \mathbb{R}$ is given by

$$\eta_y(x) = \frac{1}{2}(y - f(x))^\top \Gamma_{\text{noise}}^{-1}(y - f(x)), \quad (3)$$

and $Z := \mathbb{E}^{\mu_0}[e^{-\eta_y}] = \int_{\mathbb{R}^d} \exp(-\eta_y(x)) d\mu_0(x) > 0$ is a normalization constant whose computation is often intractable. We seek to draw samples from the posterior distribution, whose probability density, known up to a constant, is denoted by π_y .

2.2. Stein variational methods

Typically, sampling from the prior distribution μ_0 is tractable, while sampling from the posterior distribution

μ_y is a great challenge. One method to sample from the posterior is to find a transport map $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ that pushes forward the prior distribution to the posterior distribution by minimizing the Kullback–Leibler divergence

$$\mathcal{D}_{\text{KL}}(T_*\mu_0|\mu_y) := \int_{\mathbb{R}^d} \log \left(\frac{dT_*\mu_0}{d\mu_y}(x) \right) dT_*\mu_0(x), \quad (4)$$

where $T_*\mu_0$ represents the pushforward measure such that

$$\int_{\mathbb{R}^d} g(x) dT_*\mu_0(x) = \int_{\mathbb{R}^d} g \circ T(x) d\mu_0(x) \quad (5)$$

for any $T_*\mu_0$ -measurable function g . The Stein variational methods (Liu & Wang, 2016; Detommaso et al., 2018) simplify minimization of (4) for one possibly very complex and nonlinear transport map T to a sequence of simpler transport maps that are perturbations of the Identity, i.e., $T = T_L \circ T_{L-1} \circ \dots \circ T_2 \circ T_1$, $L \in \mathbb{N}$, where

$$T_l(x) = \text{Id}(x) + \varepsilon P_l(x), \quad l = 1, \dots, L, \quad (6)$$

with $\text{Id}(x) = x$, step size ε , perturbation map P_l . Let μ_l denote the pushforward measure $\mu_l := (T_l \circ \dots \circ T_1)_*\mu_0$, and let the cost functional $\mathcal{J}_l(P)$ be defined as

$$\mathcal{J}_l(P) := \mathcal{D}_{\text{KL}}((\text{Id} + P)_*\mu_{l-1}|\mu_y), \quad (7)$$

then at step l , Stein variational methods provide

$$P_l = -\mathcal{H}_l^{-1} \nabla \mathcal{J}_l(0) \quad (8)$$

where $\nabla \mathcal{J}_l(0) \in \mathbb{R}^d$ is the Fréchet derivative of $\mathcal{J}_l(P)$ evaluated at $P = 0$, $\mathcal{H}_l \in \mathbb{R}^{d \times d}$ is a rescaling matrix. For the SVGD method (Liu & Wang, 2016), $\mathcal{H}_l = \text{Id}$, while for the SVN method (Detommaso et al., 2018), $\mathcal{H}_l \approx \nabla^2 \mathcal{J}_l(0)$, an approximation of the Hessian of the cost functional $\nabla^2 \mathcal{J}_l(0)$.

By an ansatz representation of the perturbation map P_l as

$$P_l(x) = \sum_{n=1}^N k_n(x) c_n, \quad (9)$$

where $c_n \in \mathbb{R}^d$ are coefficient vectors and $k_n(x) \in \mathbb{R}$ are the basis functions, $n = 1, \dots, N$, which are shown in (Detommaso et al., 2018) to satisfy

$$\mathbb{H}c = -g, \quad (10)$$

where $c = (c_1^\top, \dots, c_N^\top)^\top \in \mathbb{R}^{Nd}$ is the coefficient vector, $g = (g_1^\top, \dots, g_N^\top)^\top \in \mathbb{R}^{Nd}$ is the gradient vector with

$$g_m := \mathbb{E}^{\mu_l}[-\nabla_x \log(\pi_y) k_m - \nabla_x k_m], \quad (11)$$

for $m = 1, \dots, N$, where ∇_x denotes the gradient operator with respect to the parameter x . \mathbb{H} is the Hessian matrix. It is specified as Identity in the SVGD method (Liu & Wang, 2016), which leads to $c_n = -g_n$, $n = 1, \dots, N$. In

the SVN method (Detommaso et al., 2018), an entry \mathbb{H}_{mn} , $m, n = 1, \dots, N$, is given by

$$\mathbb{H}_{mn} := \mathbb{E}^{\mu_l} [-\nabla_x^2 \log(\pi_y) k_n k_m + \nabla_x k_n (\nabla_x k_m)^\top], \quad (12)$$

with Hessian operator ∇_x^2 . To solve the coupled $Nd \times Nd$ system (10), a diagonal approximation is used in (Detommaso et al., 2018), i.e.,

$$\mathbb{H}_{mm} c_m = -g_m, \quad m = 1, \dots, N. \quad (13)$$

At every step l , the expectation $\mathbb{E}^{\mu_l}[\cdot]$ in (11) and (12) is evaluated by sample average approximation at the current samples x_1, \dots, x_N , which are moved according to (6) once the coefficients c_1, \dots, c_N are obtained. We remark that in the original SVGD method (Liu & Wang, 2016), the samples are moved with the simplified perturbation $P_l(x_m) = c_m$.

In both (Liu & Wang, 2016) and (Detommaso et al., 2018), the basis functions $k_n(x)$ are specified as $k_n(x) = k(x, x')$ at $x' = x_n$, where $k(x, x')$ is a suitable kernel function, e.g., a Gaussian kernel given by

$$k(x, x') = \exp\left(-\frac{1}{2}(x - x')^\top M(x - x')\right), \quad (14)$$

where M is a metric that measures the distance between x and $x' \in \mathbb{R}^d$. In (Liu & Wang, 2016), it is specified as rescaled identity matrix αId for $\alpha > 0$ possibly depending on the samples, while in (Detommaso et al., 2018), M is given by $M = \mathbb{E}^{\mu_l} [-\nabla_x^2 \log(\pi_y)]/d$ to account for the geometry of the posterior distribution by the averaged Hessian information of its density. This was shown to accelerate convergence for both SVGD and SVN compared to αId .

3. Projected Stein variational Newton

In this section, we present a projected SVN method to address high-dimensional Bayesian inference problems.

3.1. Hessian-based subspace

Without loss of generality, we consider a Gaussian prior $\mu_0 = \mathcal{N}(\bar{x}, C_{\text{pr}})$ with mean \bar{x} and covariance $C_{\text{pr}} \in \mathbb{R}^{d \times d}$, which is symmetric, positive, and definite. For linear Bayesian inference problem, the posterior is also Gaussian given by $\mu_y = \mathcal{N}(x_{\text{MAP}}, C_{\text{post}})$, where (Stuart, 2010)

$$C_{\text{post}}^{-1} = \nabla_x^2 \eta_y + C_{\text{pr}}^{-1}, \quad x_{\text{MAP}} = \bar{x} - C_{\text{post}} \nabla_x \eta_y(\bar{x}). \quad (15)$$

It is well known that the eigenvalues λ_i , $i = 1, 2, \dots$, of the generalized Hermetian eigenvalue problem

$$\nabla_x^2 \eta_y \psi_i = \lambda_i C_{\text{pr}}^{-1} \psi_i, \quad \text{where } \psi_i^\top C_{\text{pr}}^{-1} \psi_j = \triangle_{ij}, \quad (16)$$

measures the relative variation in directions ψ_i between the data-dependent likelihood and the prior. For $\lambda_i \ll 1$, the

data result in negligible variation compared to the prior in direction ψ_i , or in another words, the data provides negligible information in direction ψ_i . For general nonlinear Bayesian inference problems where the posterior is not necessarily Gaussian, the data-informed directions can be similarly obtained by the eigenvectors corresponding to the largest eigenvalues of the generalized Hermetian eigenvalue problem,

$$H \psi_i = \lambda_i C_{\text{pr}}^{-1} \psi_i, \quad \text{where } \psi_i^\top C_{\text{pr}}^{-1} \psi_j = \triangle_{ij}, \quad (17)$$

where H can be taken as an averaged Hessian (Cui et al., 2016)

$$H = \mathbb{E}^{\mu_y} [\nabla_x^2 \eta_y] \approx \frac{1}{N} \sum_{n=1}^N \nabla_x^2 \eta_y(x_n), \quad (18)$$

or a combined Hessian (Chen & Ghattas, 2018) to account for the variation of the Hessian at different samples x_n , $n = 1, \dots, N$, from the posterior distribution. Let $(\lambda_i, \psi_i)_{1 \leq i \leq r}$ denote the r largest eigenpairs such that $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_r| \geq \varepsilon_\lambda > |\lambda_{r+1}|$ for some small $\varepsilon_\lambda < 1$. Then the Hessian-based subspace spanned by the eigenvectors $\Psi = (\psi_1, \dots, \psi_r)$ captures the most variation of the parameter x informed by data y .

We remark that to solve the generalized Hermetian eigenvalue problem (17), we employ a randomized SVD algorithm (Halko et al., 2011), which requires $O(NrC_h + dr^2)$ flops, where C_h is the cost for Hessian action in one direction.

3.2. Parameter and distribution projection

For the parameter x following the prior distribution $\mu_0 = \mathcal{N}(\bar{x}, C_{\text{pr}})$, we project $x - \bar{x}$ into the Hessian-based subspace spanned by the eigenvectors Ψ as

$$x^r - \bar{x} = \sum_{i=1}^r \psi_i (\psi_i^\top C_{\text{pr}}^{-1} (x - \bar{x})) = \Psi w, \quad (19)$$

where $w = (w_1, \dots, w_r)^\top \in \mathbb{R}^r$ with $w_i = \psi_i^\top C_{\text{pr}}^{-1} (x - \bar{x})$. By $x \sim \mu_0^r := \mathcal{N}(\bar{x}, C_{\text{pr}})$, we have $w \sim \mu_0^w = \mathcal{N}(0, I_r)$ with the identity $\text{Id}_r \in \mathbb{R}^{r \times r}$, and $x^r \sim \mathcal{N}(\bar{x}, \Psi \Psi^\top)$. Let $x^\perp := x - x^r$. We have that x^\perp lives in the complement subspace $\Psi_\perp := \text{Id} - \Psi \Psi^\top C_{\text{pr}}^{-1}$, which is orthogonal to Ψ since $\Psi_\perp \Psi = 0$. Moreover, we have $x^\perp \sim \mu_0^\perp := \mathcal{N}(0, C_{\text{pr}} - \Psi \Psi^\top)$, which is independent of x^r . Therefore, the prior distribution can be decomposed for $x = x^r + x^\perp$ as

$$\mu_0(dx) = \mu_0^r(dx^r) \mu_0^\perp(dx^\perp). \quad (20)$$

Under the assumption that the variation of x^\perp informed by data y is negligible, we can freeze the distribution of x^\perp and only update the distribution of x^r by data y from its prior to

its posterior μ_y^r in the image of Ψ by the Bayes rule as

$$\frac{d\mu_y^r}{d\mu_0^r}(x^r) = \frac{1}{Z_r} \exp(-\eta_y(x^r)), \quad (21)$$

where $Z_r := \mathbb{E}^{\mu_0^r}[e^{-\eta_y}]$. Then the posterior distribution of x can be approximated by the product measure

$$\mu_y(dx) \approx \mu_y^r(dx^r) \mu_0^\perp(dx^\perp). \quad (22)$$

We remark that as $r \rightarrow \infty$, or the subspace Ψ includes all the directions that are informed by data y , the product measure in (22) converges to the true posterior distribution.

Because the uncertainty of x^r is fully represented by w through the projection (19), and the distribution update of x^r is in the image of Ψ , instead of the distribution update for x^r via (21), we can update the distribution of w by data y from its prior μ_0^w to the posterior μ_y^w by the Bayes rule

$$\frac{d\mu_y^w}{d\mu_0^w}(w) = \frac{1}{Z_w} \exp(-\eta_y(w)), \quad (23)$$

where $\eta_y^w(w) := \eta_y(x^r + x^\perp) = \eta_y(\Psi w + \bar{x} + x^\perp)$, and $Z_w := \mathbb{E}^{\mu_0^w}[e^{-\eta_y^w}]$. Therefore, to draw a sample x from its posterior distribution given by (2), we can first draw a sample w from its prior, and perform the projection (19) to obtain x^r and x^\perp . By pushing the projected prior sample w to its posterior correspondence w^y , e.g., via a transport map, we can reconstruct a posterior sample $x^y = \Psi w^y + \bar{x} + x^\perp$.

We remark that the dimension of w is r , which is often much smaller than the full dimension d of x when d is large. Moreover, this r typically does not change when d increases beyond a critical value, as we can observe from the numerical experiments in Section 4. In principle, drawing samples of the low-dimensional parameter w from its posterior distribution given by (23) is computationally faster in terms of convergence than drawing samples of the high-dimensional parameter x from its posterior distribution given by (2).

3.3. Projected Stein variational Newton

To draw samples of w from its posterior distribution given by (23), we can now use the Stein variational methods presented in the subsection 2.2. In particular, since we have used the Hessian information to construct the subspace, it is natural to also use it in the SVN method. By the same derivation of the SVN, at $l = 1, 2, \dots$, we seek a transport map

$$T_l^w(w) = I(w) + \varepsilon P_l^w(w), \quad l = 1, \dots, L, \quad (24)$$

where the perturbation map P_l^w for w is represented as

$$P_l^w(w) = \sum_{n=1}^N k_n^w(w) c_n^w, \quad (25)$$

where the coefficient vector $c^w = ((c_1^w)^\top, \dots, (c_N^w)^\top)^\top \in \mathbb{R}^{Nr}$ is the solution of the linear system

$$\mathbb{H}^w c^w = -g^w. \quad (26)$$

Here the gradient g^w is defined with its m -th entry as

$$g_m^w := \mathbb{E}^{\mu^w} [-\nabla_w \log(\pi_y^w) k_m^w - \nabla_w k_m^w], \quad (27)$$

where π_y^w denote the posterior density of w , and the Hessian \mathbb{H}^w is defined with its mn -th entry as

$$\mathbb{H}_{mn}^w := \mathbb{E}^{\mu^w} [-\nabla_w^2 \log(\pi_y^w) k_n^w k_m^w + \nabla_w k_n^w (\nabla_w k_m^w)^\top]. \quad (28)$$

By the definition of the projection (19), we have

$$\nabla_w \log(\pi_y^w(w)) = \Psi^\top \nabla_x \log(\pi_y(x^r + x^\perp)), \quad (29)$$

and

$$\nabla_w^2 \log(\pi_y^w(w)) = \Psi^\top \nabla_x^2 \log(\pi_y(x^r + x^\perp)) \Psi. \quad (30)$$

We use a Gaussian kernel k_n^w for the ansatz (25) as

$$k_n^w(w) = \exp\left(-\frac{1}{2}(w - w_n)^\top M^w (w - w_n)\right), \quad (31)$$

where the metric M^w is given by an averaged Hessian

$$M^w = -\frac{1}{r} \frac{1}{N} \sum_{n=1}^N \nabla_w^2 \log(\pi_y^w(w_n)). \quad (32)$$

The gradient of the kernel k_n^w is given by

$$\nabla_w k_n^w(w) = k_n^w(w) M^w (w_n - w). \quad (33)$$

We remark that the projected system (26) is of size $Nr \times Nr$, which is a considerable reduction from the full system (10) of size $Nd \times Nd$ as $r \ll d$. To further reduce the size of the coupled system (26), we employ a ‘‘mass-lumping’’ technique to decouple it as N systems of size $r \times r$ as

$$\mathbb{H}_m^w c_m^w = -g_m^w, \quad m = 1, \dots, N. \quad (34)$$

where g_m^w is given as in (29), while \mathbb{H}_m^w is given by

$$\mathbb{H}_m^w := \left(\sum_{n=1}^N \mathbb{H}_{mn}^w \right), \quad m = 1, \dots, N. \quad (35)$$

where \mathbb{H}_{mn}^w are defined in (30).

3.4. Globalization by line search

Except for in the case of a linear inference problem, the cost functional—Kullback–Leibler divergence—is nonconvex. In the case of that the Newton approximation to the Kullback–Leibler divergence is locally exact, the simple choice of $\varepsilon = 1$ is the optimal choice for the step size.

However, since the geometry generally exhibits complex non-quadratic local structure, a constant stepsize ε renders minimization of \mathcal{D}_{KL} inefficient. A careful choice of the step size ε is crucial for both fast convergence and stability of Stein variational methods. While, there are many options to choose from, we employ an Armijo line search globalization method to choose this step size, to much success. Specifically, at step $l = 1, 2, \dots$, we seek ε to minimize the Kullback–Leibler divergence

$$\mathcal{D}_{\text{KL}}((T_l^w)_* \mu_{l-1}^w | \mu_y^w) = \mathcal{D}_{\text{KL}}(\mu_{l-1}^w | (T_l^w)^* \mu_y^w), \quad (36)$$

where $(T_l^w)^*$ is the pullback operator. Because

$$\begin{aligned} \mathcal{D}_{\text{KL}}(\mu_{l-1}^w | (T_l^w)^* \mu_y^w) &= \mathbb{E}^{\mu_{l-1}^w} [\log(\pi_{l-1}^w(\cdot))] \\ &\quad - \mathbb{E}^{\mu_{l-1}^w} [\log(\pi_y(T_l^w(\cdot)) |\det \nabla_w T_l^w(\cdot)|)], \end{aligned} \quad (37)$$

where the first term, in which π_{l-1}^w denotes the density function of the distribution μ_{l-1}^w , does not depend on ε . Hence we only need to consider the second term denoted as $\mathcal{D}_{\text{KL}}^{(2)}$, which is evaluated by the sample average approximation as

$$\begin{aligned} \mathcal{D}_{\text{KL}}^{(2)} &\approx -\frac{1}{N} \sum_{n=1}^N \log(\pi_y(T_l^w(w_n^{l-1}))) \\ &\quad -\frac{1}{N} \sum_{n=1}^N \log(|\det \nabla_w T_l^w(w_n^{l-1})|), \end{aligned} \quad (38)$$

which can be readily computed for every ε . We remark that the second term of (38) is close to 0 when the kernel function $k_n^w(w)$ in (25) is close to 0 at every sample w_m^{l-1} for $m \neq n$, so we only need to consider the first term of (38). Moreover, to guarantee that $\mathcal{D}_{\text{KL}}^{(2)}$ is reduced for a suitable ε , we can find sample-dependent step sizes $\varepsilon(w_n^{l-1})$ such that

$$-\log(\pi_y(T_l^w(w_n^{l-1}))) \quad (39)$$

is reduced for each $n = 1, \dots, N$, which provides great flexibility and stability for the Stein variational methods.

3.5. A two-level adaptive pSVN algorithm

To this end, given the bases Ψ as the data-informed parameter directions, we can draw samples x_1, \dots, x_N from the prior distribution and drive them by projected SVN to match the posterior distribution in a low-dimensional subspace, while keeping the components of the samples in the complement subspace unchanged. We summarize the one-level (in contrast to a two-level algorithm presented later) projected SVN method in Algorithm 1, in which the stopping criterion could be set as one or a combination of the following:

1. the maximum norm of the updates $w_m^l - w_m^{l-1}$, $m = 1, \dots, N$, is smaller than a given tolerance To_g ;

2. the maximum norm of the gradients g_m , $m = 1, \dots, N$, is smaller than a given tolerance To_w ;
3. the number of iterations l reaches a preset number L .

Algorithm 1 pSVN

- 1: **Input:** prior samples x_1, \dots, x_N , bases Ψ , density π_y .
 - 2: **Output:** posterior samples x_1^y, \dots, x_N^y .
 - 3: Perform projection (19) to get the decomposition $x_n = x_n^r + x_n^\perp$ and the samples w_n^{l-1} , $n = 1, \dots, N$, at $l = 1$.
 - 4: **repeat**
 - 5: Compute the gradient and Hessian by (29) and (30).
 - 6: Compute the kernel and its gradient by (31) and (33).
 - 7: Assemble and solve system (34) for c_1^w, \dots, c_N^w .
 - 8: Perform a line search by (39) to get w_1^l, \dots, w_N^l .
 - 9: Update the samples $x_n^r = \Psi w_n^l + \bar{x}$, $n = 1, \dots, N$.
 - 10: Set $l \leftarrow l + 1$.
 - 11: **until** A stopping criterion is met.
 - 12: Reconstruct samples $x_n^y = x_n^r + x_n^\perp$, $n = 1, \dots, N$.
-

In Algorithm 1, we assume that the bases Ψ for the projection are the data informed parameter directions, which are obtained by the Hessian-based algorithm in Section 3.1 at the posterior samples x_1, \dots, x_N . However, we do not have these samples but only the prior samples at the beginning. To address this problem, we propose a two-level adaptive algorithm that adaptively construct the bases Ψ along the push of the prior samples to the posterior samples. This is presented in Algorithm 2. We remark that the same stopping criteria in Algorithm 2 as those in Algorithm 1 are used with smaller tolerances $\text{To}_g^2, \text{To}_w^2$ for the gradients and the updates, e.g., $\text{To}_g^2 = 10^{-1} \text{To}_g$, and $\text{To}_w^2 = 10^{-1} \text{To}_w$.

Algorithm 2 Adaptive pSVN

- 1: **Input:** prior samples x_1, \dots, x_N , density π_y .
 - 2: **Output:** posterior samples x_1^y, \dots, x_N^y .
 - 3: Set level $l_2 = 1$, $x_n^{l_2-1} = x_n$, $n = 1, \dots, N$.
 - 4: **repeat**
 - 5: Perform the eigendecomposition (17) at samples $x_1^{l_2-1}, \dots, x_N^{l_2-1}$, and form the bases Ψ^{l_2} .
 - 6: Apply **Algorithm 1** of pSVN to update the samples $[x_1^{l_2}, \dots, x_N^{l_2}] = \text{pSVN}([x_1^{l_2-1}, \dots, x_N^{l_2-1}], \Psi^{l_2}, \pi_y)$.
 - 7: Set $l_2 \leftarrow l_2 + 1$.
 - 8: **until** A stopping criterion is met.
-

3.6. Parallel computation and implementation

In this section, we take advantage of the projected SVN in low-dimensional subspaces—including fast computation, light communication, and a low memory consumption—and present an efficient and parallel implementation.

We present a parallel pSVN using MPI communication in Algorithm 3. In particular, lines 4, 7, 9, 12 manage

the data communication and all other lines perform local computation in each processor core. Lines 4 and 12 involve global communication (gather and broadcast) of the low-dimensional samples w_m , $m = 1, \dots, M$, of size Mr , which are used for the kernel and its gradient evaluations in (31) and (33) at all samples, as well as for the sample update in (25). Line 7 involves global communication (gathers and broadcasts) of the gradients (of size Mr) and Hessians (of size Mr^2) of the log posterior density (29) and (30), which are used in the expectation evaluation at all samples for assembling the system (34). Line 9 involves global communication (gathers and broadcasts) of the kernel values (of size NM) at all samples, which are used in moving the samples by (25). Meanwhile, Line 9 gathers a local sum of the kernel values $\sum_m k_m(w)$ (of size N) and its gradients $\sum_m \nabla_w k_m(w)$ (of size rN), performs a global sum of them, and broadcasts the results to all cores, which are used for assembling the lumped Hessian (35). In summary, the data volumes of communication in Algorithm 3 are bounded by $\max(Mr^2, MN)$ floats.

Algorithm 3 Parallel pSVN using MPI

- 1: **Input:** M prior samples, x_1, \dots, x_M , in each of K cores, bases Ψ , and density π_y in all cores.
 - 2: **Output:** posterior samples x_1^y, \dots, x_M^y in each core.
 - 3: Perform projection (19) to get $x_m = x_m^r + x_m^\perp$ and the samples w_m^{l-1} , $m = 1, \dots, M$, at $l = 1$.
 - 4: Perform MPI Allgather for w_m^{l-1} , $m = 1, \dots, M$.
 - 5: **repeat**
 - 6: Compute the gradient and Hessian by (29) and (30).
 - 7: Perform MPI Allgather for the gradient and Hessian.
 - 8: Compute the kernel and its gradient by (31) and (33).
 - 9: Perform MPI Allgather for k_m , $m = 1, \dots, M$, MPI Allreduce w. sum for $\sum_m k_m^w$ and $\sum_m \nabla_w k_m^w$.
 - 10: Assemble and solve system (34) for c_1^w, \dots, c_M^w .
 - 11: Perform a line search by (39) to get w_1^l, \dots, w_M^l .
 - 12: Perform MPI Allgather for w_m^l , $m = 1, \dots, M$.
 - 13: Update the samples $x_m^r = \Psi w_m^l + \bar{x}$, $m = 1, \dots, M$.
 - 14: Set $l \leftarrow l + 1$.
 - 15: **until** A stopping criterion is met.
 - 16: Reconstruct samples $x_m^y = x_m^r + x_m^\perp$, $m = 1, \dots, M$.
-

To implement a parallel version of the adaptive pSVN Algorithm 2, we only need to construct the bases Ψ in parallel to replace its Line 5, for which we perform an averaged Hessian action in random directions with M samples in each core by $O(M(rC_h))$ flops, followed by a MPI Allreduce with a SUM operator to get a global averaged Hessian action (18) before performing randomized SVD with $O(dr^2)$ flops. The data volumes for communication is dr floats, which dominates all other communication cost if d is so large that $dr > \max(r^2M, NM)$. Alternatively, we can construct the bases Ψ using Hessian at the local samples in each core without communication for Ψ .

4. Numerical experiments

We demonstrate the convergence, accuracy, and scalability of the pSVN method by three models. We consider two inference problems constrained by partial differential equations (PDE), one linear problem with Gaussian posterior to demonstrate the convergence and accuracy of pSVN in comparison with SVN and SVGD, the other nonlinear problem to demonstrate the scalability of pSVN. For the latter purpose, we also consider a Bayesian autoencoder problem. The code is available at our Bitbucket repository (pSV) for all the three test problems.

4.1. A linear inference problem

The linear inference problem is constrained by the PDE

$$-\Delta u + u = x, \text{ in } (0, 1), \quad u(0) = 0, \quad u(1) = 1. \quad (40)$$

15 pointwise observations of u with 1% noise are uniformly distributed in $(0, 1)$. The input x is a random field with Gaussian prior $\mathcal{N}(0, C)$, where $C = (I - 0.1\Delta)^{-1}$ with identity Id and Laplace operator Δ . The posterior of x is also Gaussian given as in (15). We solve this forward model by a finite element method with piecewise elements on a uniform mesh of size 2^{10} , which leads to 1025 dimensions of the discretized parameter x .

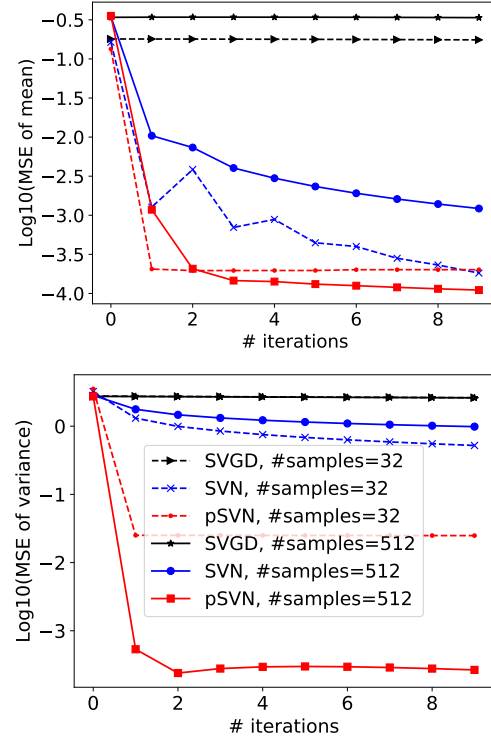


Figure 1. Decay of the mean squared errors (with 10 trials) of the L2-norm of the mean (top) and pointwise variance (bottom) of the parameter using 32 and 512 samples by SVGD, SVN, and pSVN.

The L2-norm of the mean and pointwise variance of the

parameter x w.r.t. its posterior distribution can be explicitly computed using (15), which serve as the reference for the sample approximation errors. Figure 1 compares the convergence and accuracy of SVGD, SVN, and pSVN by the decay of the mean squared errors (using 10 trials) of the above two quantities. We can observe a much faster convergence and higher accuracy of pSVN than SVGD and SVN, for both mean and especially variance that measures the goodness of samples. In fact, pSVN converges with just one iteration in a subspace of dimension 5 (at $\varepsilon_\lambda = 0.1$ in Section 3.1), while the convergence of the variance by SVN is extremely slow because of the high-dimensionality. Moreover, we can see that the convergence of SVN becomes slower for increasing number of samples. With the same number of iterations of SVN and pSVN, the SVGD method produces no evident decay of the approximation errors.

4.2. A nonlinear inference problem

The nonlinear inference problem is constrained by the PDE

$$-\nabla \cdot (e^x \nabla u) = 0, \text{ in } (0, 1)^2, \quad (41)$$

with Dirichlet boundary conditions on the top ($u = 1$) and bottom ($u = 0$) boundaries, and zero Neumann boundary conditions on the left and right boundaries. 49 pointwise observations of u with 1% noise are uniformly distributed in $(0, 1)^2$. The input x is a random field with Gaussian prior $\mathcal{N}(0, C)$, where $C = (I - 0.1\Delta)^{-2}$. We solve this forward model by a finite element method with piecewise elements on a uniform mesh of varying sizes, which leads to a sequence of dimensions for the discretized parameter.

We focus on the demonstration of the scalability of pSVN w.r.t. the number of parameter dimensions, samples, and processor cores. Firstly, the dimension of the Hessian-based subspace r , which determines the computation and communication cost of pSVN, depends on the decay of the absolute eigenvalues $|\lambda_i|$ as presented in Section 3.1. The top part of Figure 2 shows that with increasing d , r does not change, which implies that pSVN is scalable w.r.t. the number of parameter dimensions.

Secondly, as shown in the middle part of Figure 2, with increasing number of samples for a fixed parameter dimension $d = 1,089$, the averaged norm of the update $w^l - w^{l-1}$, as one convergence indicator presented in the subsection 3.5, decays similarly, which demonstrates the scalability of pSVN w.r.t. the number of samples.

Thirdly, in the bottom part of Figure 2 we plot the *total* wall clock time of pSVN and the time for its computational components of Algorithm 3 using different number of processor cores for the same work, i.e., the same number of samples (256), including *variation* for forward model solve, gradient and Hessian evaluation, as well as eigendecomposition, *kernel* for kernel and its gradient evaluation, *solve* for solving

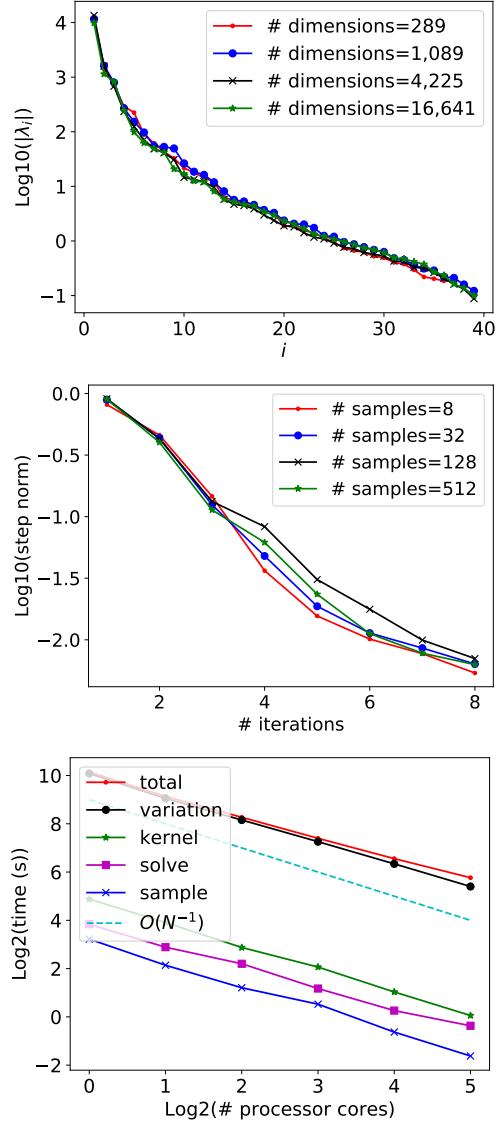


Figure 2. Top: Decay of eigenvalues with increasing dimension d . Middle: Decay of the averaged norm of the update $w^l - w^{l-1}$ w.r.t. the iteration number l , with increasing number of samples. Bottom: Decay of the wall clock time (seconds) of different computational components w.r.t. increasing number of processor cores.

the Newton system (34), and *sample* for sample projection and reconstruction. We can observe nearly perfect strong scaling w.r.t. increasing number of processor cores. Moreover, the time for *variation*, which depends on parameter dimension d , dominates the time for all other components, in particular *kernel* and *solve* whose cost only depends on r , not d .

We remark that without proper line search as introduced in the subsection 3.4 that generally guarantees a convergence of pSVN, the samples could be updated to regions that cause instability, e.g., the diffusion coefficient e^x in (41) becomes such that the PDE model is ill-posed, leading to solver crash as observed in this experiment.

4.3. A Bayesian Autoencoder Problem

We consider a Bayesian inference problem constrained by a convolutional autoencoder neural network, specifically with applications to machine learning.

In the Bayesian autoencoder problem, we seek to learn a low dimensional representation of data under uncertainty. Given input data $z \in \mathbb{R}^{\text{data}}$ the $2m$ layer autoencoder mapping is defined as

$$y(\cdot) = \circ_{i=1}^{2m} \phi_i(w_i * (\cdot) + b_i) \quad (42)$$

where w_i is the convolution kernel (weights) for layer i , and ϕ_i is an nonlinear activation functions. The $*$ operations represents both convolution and downsampling. The first m compositions map down to a low dimensional latent representation of the input data z , the last m compositions map the data back to \mathbb{R}^{data} . The target data has 5% i.i.d. noise added to it based on min-max normalization of the data. The data z for the problem are 500 randomly selected MNIST images. The inference parameter $\{x_i\} = \{(w_i, b_i)\} \in \mathbb{R}^d$ has the i.i.d. prior $\mathcal{N}(0, \sigma_i^2)$, as is common in popular weight prior techniques such as Xavier initialization—The first layer variance is set to unity and subsequent layers decay by a constant 0.5 multiplicative factor. We use a fixed convolution kernel support of 4×4 and vary the number of filters on each layer from 2, 3, 4 and use $m = 2, 4$ layers (so 4, 8 total layers). Due to the low dimensional nature of the autoencoder and the fixed data, the pSVN algorithm can efficiently find a r dimensional Hessian subspace. The dimensionality of this subspace depends on the decay of the absolute eigenvalues $|\lambda_i|$. The top plot in Figure (3) shows that the rank structure does not change drastically as the dimension d increases. The 67 dimensional problem corresponds to a $2m = 4$ layer autoencoder with 2 convolution kernels for each layer, while the 2,527 dimensional problem corresponds to a $2m = 8$ layer autoencoder with 3 convolution kernels per layer.

The pSVN algorithm is scalable not just with respect to the overall dimension d but also the number of samples drawn. For a fixed problem the gradient norm decays uniformly for various numbers of samples. With more samples however, the KL divergence is more faithfully represented, as seen in the middle plot in Figure (3).

The pSVN algorithm scales strongly with problem dimension for the Bayesian autoencoder problem, as with the case of the PDE problems the variation time is the dominant cost for the algorithm. See the bottom plot in Figure (3)

5. Conclusion

We presented a fast and scalable variational method, pSVN, for Bayesian inference in high dimensions and for problems with expensive-to-evaluate likelihoods. The method exploits the geometric structure and smoothness of the posterior via

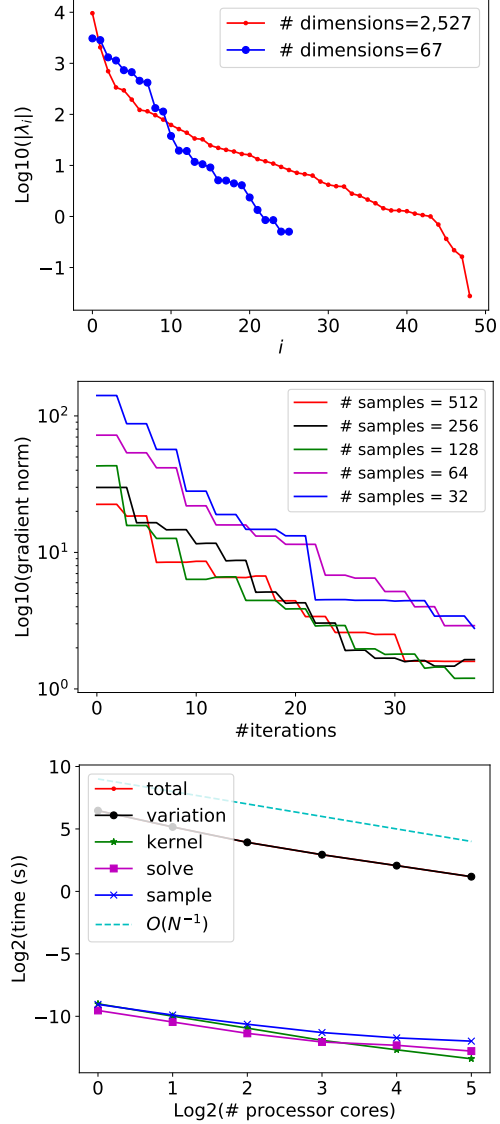


Figure 3. Top: Decay of eigenvalues for $m = 4, 8$ layer autoencoders. Middle: Gradient norm vs iterations for different sample numbers for $m = 4$ layer autoencoder. Bottom: Decay of the wall clock time (seconds) of different computational components w.r.t. increasing number of processor cores.

its Hessian operator, and the intrinsic low-dimensionality of the change from prior to posterior characteristic of many high-dimensional inference problems via low rank approximation of the (prior preconditioned) Hessian of the log likelihood, computed efficiently using randomized matrix-free SVD. The fast convergence of pSVN relative to SVGD and SVN and its scalability with respect to the parameter dimension and number of samples and processor cores were demonstrated for both linear and nonlinear inference problems.

References

- Projection Stein variational Newton. URL https://bitbucket.org/peng_ices/hippylib-stein/src/projection/.
- Bashir, O., Willcox, K., Ghattas, O., van Bloemen Waanders, B., and Hill, J. Hessian-based model reduction for large-scale systems with initial condition inputs. *International Journal for Numerical Methods in Engineering*, 73:844–868, 2008.
- Beskos, A., Girolami, M., Lan, S., Farrell, P. E., and Stuart, A. M. Geometric mcmc for infinite-dimensional inverse problems. *Journal of Computational Physics*, 335:327 – 351, 2017. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2016.12.041>. URL <http://www.sciencedirect.com/science/article/pii/S0021999116307033>.
- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.
- Bui-Thanh, T. and Ghattas, O. Analysis of the Hessian for inverse scattering problems: I. Inverse shape scattering of acoustic waves. *Inverse Problems*, 28(5):055001, 2012.
- Bui-Thanh, T., Ghattas, O., Martin, J., and Stadler, G. A computational framework for infinite-dimensional bayesian inverse problems part I: The linearized case, with application to global seismic inversion. *SIAM Journal on Scientific Computing*, 35(6):A2494–A2523, 2013.
- Chen, P. and Ghattas, O. Hessian-based sampling for high-dimensional model reduction. *arXiv preprint arXiv:1809.10255*, 2018.
- Chen, P. and Schwab, C. Sparse-grid, reduced-basis Bayesian inversion. *Computer Methods in Applied Mechanics and Engineering*, 297:84 – 115, 2015.
- Chen, P., Villa, U., and Ghattas, O. Hessian-based adaptive sparse quadrature for infinite-dimensional Bayesian inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 327:147–172, 2017.
- Chen, P., Villa, U., and Ghattas, O. Taylor approximation and variance reduction for PDE-constrained optimal control problems under uncertainty. *Journal of Computational Physics*, 2019. To appear.
- Chen, W. Y., Mackey, L., Gorham, J., Briol, F.-X., and Oates, C. J. Stein points. *arXiv preprint arXiv:1803.10161*, 2018.
- Cui, T., Law, K. J., and Marzouk, Y. M. Dimension-independent likelihood-informed MCMC. *Journal of Computational Physics*, 304:109–137, 2016.
- Detommaso, G., Cui, T., Marzouk, Y., Spantini, A., and Scheichl, R. A stein variational Newton method. In *Advances in Neural Information Processing Systems*, pp. 9187–9197, 2018.
- Girolami, M. and Calderhead, B. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.
- Halko, N., Martinsson, P.-G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
- Isaac, T., Petra, N., Stadler, G., and Ghattas, O. Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet. *Journal of Computational Physics*, 296:348–368, September 2015. doi: 10.1016/j.jcp.2015.04.047.
- Liu, C. and Zhu, J. Riemannian Stein variational gradient descent for Bayesian inference. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pp. 2378–2386, 2016.
- Martin, J., Wilcox, L., Burstedde, C., and Ghattas, O. A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing*, 34(3): A1460–A1487, 2012.
- Marzouk, Y., Moselhy, T., Parno, M., and Spantini, A. Sampling via measure transport: An introduction. In *Handbook of Uncertainty Quantification*, pp. 1–41. Springer, 2016.
- Petra, N., Martin, J., Stadler, G., and Ghattas, O. A computational framework for infinite-dimensional Bayesian inverse problems, Part II: Stochastic Newton MCMC with application to ice sheet flow inverse problems. *SIAM J. Sci. Comput.*, 36(4):A1525–A1555, 2014. ISSN 1064-8275. doi: 10.1137/130934805. URL <http://dx.doi.org/10.1137/130934805>.
- Schillings, C. and Schwab, C. Sparse, adaptive Smolyak quadratures for Bayesian inverse problems. *Inverse Problems*, 29(6):065011, 2013.

- Schillings, C. and Schwab, C. Scaling limits in computational Bayesian inversion. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50(6):1825–1856, 2016.
- Schwab, C. and Stuart, A. Sparse deterministic approximation of Bayesian inverse problems. *Inverse Problems*, 28(4):045003, 2012.
- Spantini, A., Solonen, A., Cui, T., Martin, J., Tenorio, L., and Marzouk, Y. Optimal low-rank approximations of Bayesian linear inverse problems. *SIAM Journal on Scientific Computing*, 37(6):A2451–A2487, 2015.
- Stuart, A. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19(1):451–559, 2010.
- Wang, D., Zeng, Z., and Liu, Q. Stein variational message passing for continuous graphical models. In *International Conference on Machine Learning*, pp. 5206–5214, 2018.