A Trust-Region Method for Graphical Stein Variational Inference

Liam Pavlovic[1] **David M. Rosen**[1]

[1]Northeastern University, Boston, Massachussets, USA

Abstract

Stein variational inference (SVI) is a sample-based approximate Bayesian inference technique that generates a sample set by jointly optimizing the samples' locations to minimize an information-theoretic measure of discrepancy with the target probability distribution. SVI thus provides a fast and significantly more sample-efficient approach to Bayesian inference than traditional (random-sampling-based) alternatives. However, the optimization techniques employed in existing SVI methods struggle to address problems in which the target distribution is high-dimensional, poorly-conditioned, or non-convex, which severely limits the range of their practical applicability. In this paper, we propose a novel trust-region optimization approach for SVI that successfully addresses each of these challenges. Our method builds upon prior work in SVI by leveraging conditional independences in the target distribution (to achieve high-dimensional scaling) and second-order information (to address poor conditioning), while additionally providing an effective adaptive step control procedure, which is essential for ensuring convergence on challenging non-convex optimization problems. Experimental results show our method achieves superior numerical performance, both in convergence rate and sample accuracy, and scales better in high-dimensional distributions, than previous SVI techniques.

1 INTRODUCTION

Drawing inferences from noisy data is a fundamental capability in artificial intelligence, machine learning, and scientific and engineering applications. Mathematically, this procedure is naturally expressed in the language of posterior Bayesian inference. Many of these inference problems can be formulated as probabilistic graphical models (PGMs), which are an effective tool for modeling joint distributions with known conditional independences among the individual variables. The conditional independence structure encoded in a PGM can greatly simplify the inference task [Koller and Friedman, 2009]. Nonetheless, exact Bayesian inference is typically computationally intractable, so in practice approximate inference methods are used instead.

One of the most common approximate Bayesian inference methods is sample-based approximation, which uses a sample set to represent the target distribution. This approximation has the benefits of simplicity, flexibility, arbitrary precision (as sample size increases), and easy empirical estimation of any statistic over the target distribution. Traditional methods for generating a sample-based approximation are based on random sampling. Common examples of random sampling algorithms include Markov chain Monte Carlo (MCMC) [Andrieu et al., 2003], nested sampling [Speagle, 2020], and Hamiltonian Monte Carlo [Betancourt, 2017]. Due to their dependence on random processes to explore the state space, these methods can be slow to converge and sample inefficient. These deficiencies become especially pronounced in high-dimensional problems.

Stein variational inference (SVI) is a more efficient alternative for generating a sample-based approximation [Liu and Wang, 2016]. In place of random sampling, SVI deterministically optimizes a set number of samples to minimize KL divergence with the reference distribution. SVI has been demonstrated to have superior sample efficiency over random sampling methods, capturing more information with fewer samples [Liu and Wang, 2016]. However, SVI can still struggle to scale to high-dimensional, ill-conditioned, and non-convex objectives. Previous work [Zhuo et al., 2018, Wang et al., 2018, Detommaso et al., 2018] on SVI methods has addressed some of these challenges individually, but no single previous SVI method for PGM problems handles all these challenges well.

In this paper, we propose a novel trust-region optimization approach for SVI that successfully addresses each of these challenges. Our method builds upon prior work by leveraging both conditional independences in the target distribution to achieve high-dimensional scaling [Zhuo et al., 2018, Wang et al., 2018] and second-order information to address poor conditioning [Detommaso et al., 2018] in the same system. We also provide an effective adaptive step control procedure for SVI, which is essential for ensuring convergence on challenging non-convex optimization problems. Experimental results show our method achieves superior numerical performance, both in convergence rate and sample accuracy, and scales better to high-dimensional distributions than previous variational inference techniques. Code for our method and experiments is available at https://github.com/NEURAL/TrustRegionSVI.

2 STEIN VARIATIONAL INFERENCE

The objective of SVI [Liu and Wang, 2016] is to approximate a given target distribution $p(x)$ on $\mathcal{X} \subseteq \mathbb{R}^D$ using the Kullback-Leibler (KL) divergence-minimizing representative q within some family Q of tractable model distributions

$$\min_{q \in Q} KL(q||p) \equiv \mathbb{E}_{x \sim q}[\log q(x) - \log p(x)] \quad (1)$$

To achieve this, we begin with some initial distribution q_0 and generate a set of pushforwards $q_1, ..., q_L$ according to the rule $q_{l+1} = (T_l)_* q_l$ where $T_l \in \mathbb{R}^D \to \mathbb{R}^D$ is some perturbation $T_l = I + \Phi_l$ of the identity map. At each iteration l, we seek a perturbation function Φ_l from some function space $\mathcal{F}$ such that

$$J[\Phi_l] \triangleq KL((I + \Phi_l)_* q_l||p) \quad J[\Phi_l] < J[\mathbf{0}] \quad (2)$$

To ensure the descent condition, we can choose Φ_l to be an infinitesimal application of the negative functional gradient at $J[\mathbf{0}]$ in $\mathcal{F}$. For a Hilbert space $\mathcal{F}$, the gradient of a functional J at some $S \in \mathcal{F}$ is defined as the element $\nabla J[S] \in \mathcal{F}$ satisfying

$$\langle \nabla J[S], V \rangle_{\mathcal{F}} = DJ[S](V) \quad \text{for all } V \in \mathcal{F}$$

$$DJ[S](V) \triangleq \lim_{\tau \to 0} \frac{1}{\tau}(J[S + \tau V] - J[S])$$

Consider some kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ with a corresponding reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. A particularly advantageous choice for $\mathcal{F}$ is the vector-valued RKHS $\mathcal{H}^D$ because this space has a closed form for the desired functional gradient $\nabla J[\mathbf{0}]$ [Liu and Wang, 2016]

$$\nabla J[\mathbf{0}](x) = -\mathbb{E}_{z \sim q}[k(z, x) \nabla_z \log p(z) + \nabla_z k(z, x)] \quad (3)$$

In order to compute the expectation above, we need some tractable representation of the distribution q. A natural choice for this representation is a *sample*, since this parameterization is both flexible, and makes the expectation in Eq. 3 trivial to approximate.

Stein variational gradient descent (SVGD) [Liu and Wang, 2016] combines a sample-based approximation of q and the descent direction in Eq. 3 into an iterative procedure for generating a sample-based approximation of p. SVGD first samples a set of points $\{x_i\}_{i=1}^n$ from an initial distribution q_0 and then iteratively updates the location of each sample using some static step size ξ and a sample-based approximation of Eq. 3

$$x_i \leftarrow x_i + \xi \frac{1}{n} \sum_{j=1}^{n} k(x_j, x_i) \nabla_{x_j} \log p(x_j) + \nabla_{x_j} k(x_j, x_i) \quad (4)$$

The first term of this update pushes the samples towards high probability areas of p while the second term, referred to as the *kernel repulsion term*, pushes apart samples that are close together. The kernel repulsion term is essential because it spreads the sample across the distribution [Liu and Wang, 2016].

3 RELATED WORK

SVGD [Liu and Wang, 2016], as discussed above, struggles to scale to high-dimensional, non-convex, and ill-conditioned objectives. Several subsequent works have suggested modifications to address these challenges.

There are two major contributors to SVI's poor performance in high dimensions. First, the information (in bits) required to encode the joint target distribution p grows exponentially with respect to it's dimension. This increases the amount of information approximation methods (like SVI) must infer to accurately approximate p [Koller and Friedman, 2009]. Second, when using distance-based kernels, like the radial basis function (RBF) kernel, the magnitude of the kernel repulsion term decreases in higher dimensions, resulting in mode collapse, where all the sample points are densely packed around a single mode of the target distribution [Zhuo et al., 2018]. When the conditional factorization of p is known (i.e. when p is represented by a PGM), these conditional independence relations can be exploited to dramatically simplify the inference task and address these challenges [Koller and Friedman, 2009].

Graphical Stein variational inference methods [Zhuo et al., 2018, Wang et al., 2018] exploit the conditional independences encoded in the PGM for the target p via their kernel design. Assuming the space $\mathcal{X}$ is a product of factors $\mathcal{X} = \mathcal{C}_1 \times ... \times \mathcal{C}_D$, graphical SVI employs a set of D local kernels $k_a : \mathcal{X}_{\mathcal{S}_a} \times \mathcal{X}_{\mathcal{S}_a} \mapsto \mathbb{R}$, where $\mathcal{S}_a$ represents the factor $\mathcal{C}_a$ and its Markov blanket Γ_a. In this local kernel setting, the Hilbert space $\mathcal{F}$ over which we take functional gradients

becomes the product $\mathcal{H}_1 \times ... \times \mathcal{H}_D$ of the local kernels' RKHSs. In this space, the closed form for the functional gradient $\nabla \hat{J}[\mathbf{0}]$ ($\wedge$ decorator used to indicate usage of local kernels and differentiate from the gradient in Eq. 3) is given by

$$(\nabla \hat{J}[\mathbf{0}](x))_a = -\mathbb{E}_{z\sim q}[k_a(z,x)\nabla_{z_a} \log p(z) + \nabla_{z_a} k_a(z,x)] \quad \text{for } a \in 1, ..., D \quad (5)$$

This functional gradient interpretation of the graphical SVI update follows naturally from previous work [Zhuo et al., 2018, Wang et al., 2018], but has not been presented before. So, we present a proof in Appendix A. Note that utilizing the above descent direction only guarantees that the resulting approximation q agrees with the target's *conditional* distributions: $q(x_a|x_{\Gamma_a}) = p(x_a|x_{\Gamma_a})$ [Zhuo et al., 2018, Wang et al., 2018], indicating that these methods inference the conditionals as desired.

Other work on SVI has explored improving high-dimensional performance on problems without conditional structure by employing Grassman manifold [Liu et al., 2022], low-dimensional subspace projection [Chen and Ghattas, 2020, Chen et al., 2019], and slicing [Gong et al., 2021] strategies.

Many previous SVI methods only utilize first-order information in their updates, which is often insufficient to achieve good convergence on ill-conditioned objectives. The Stein variational Newton method (SVN) [Detommaso et al., 2018] incorporates second-order information by deriving a Newton system to compute an approximate Newton update w_i for each sample x_i. This Newton system is block-structured

$$\begin{bmatrix} H(x_1,x_1) & \dots & H(x_1,x_n) \\ \vdots & \ddots & \vdots \\ H(x_n,x_1) & \dots & H(x_n,x_n) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} -\nabla J[\mathbf{0}](x_1) \\ \vdots \\ -\nabla J[\mathbf{0}](x_n) \end{bmatrix}$$

where the ab-th entry of the Hessian matrix block $H(x,y)$ is

$$(H(x,y))_{ab} = \mathbb{E}_{z\sim q}[-k(z,x)k(z,y)\partial_{ab} \log p(z) + \partial_{z_b} k(z,x)\partial_{z_a} k(z,y)] \quad (6)$$

By applying a block-diagonal approximation to the original system, SVN solves a decoupled system for the approximate Newton update w_i of each sample x_i

$$H(x_i,x_i)w_i = -\nabla J[\mathbf{0}](x_i) \quad \text{for } i \in 1, ..., n \quad (7)$$

To solve these decoupled systems, SVN utilizes a conjugate gradient method such as the Newton-CG method [Nocedal and Wright, 1999]. Like SVGD, SVN does not leverage conditional independence, and thus performs poorly in high dimensions.

Other prior works have leveraged second order information to improve the performance of other variational inference techniques such as stochastic gradient MCMC [Wang et al., 2021] and Langevin dynamics sampling [Wang and Li, 2020].

Importantly, the previously discussed graphical SVI methods do not utilize an adaptive method for step control, which was identified in Detommaso et al. [2018] as important future work. Adaptive step control has been implemented for low-dimensional subspace projection methods [Chen et al., 2019]. However, this method selects step sizes to minimize the KL divergence between *projections* of the target p and sample set $\{x_i\}_{i=1}^n$ onto a low-dimensional subspace, which does not necessarily guarantee a reduction of the divergence between the target p and sample set $\{x_i\}_{i=1}^n$ themselves.

4 EXPLOITING CONDITIONAL INDEPENDENCE IN SECOND-ORDER SVI

4.1 SECOND VARIATION IN LOCAL KERNEL SETTING

As a first contribution, we show how to implement a second-order Hessian model for SVI that exploits conditional independence. To do this, we generalize the formula for the second-order variation presented in SVN [Detommaso et al., 2018] to the local kernel space $\mathcal{H}_1 \times ... \times \mathcal{H}_D$ utilized by graphical SVI methods. The second variation is defined as the directional derivative along a pair of directions $V, W \in \mathcal{H}_1 \times ... \times \mathcal{H}_D$

$$D^2 J[\mathbf{0}](V,W) = \lim_{\tau \to 0} \frac{1}{\tau}(DJ[\tau W](V) - DJ[\mathbf{0}](V))$$

Theorem 1. *Along a pair of directions $V, W \in \mathcal{H}_1 \times ... \times \mathcal{H}_D$ the second variation is* [1]

$$D^2 J[\mathbf{0}](V,W) = \sum_{a=1}^{D}\sum_{b=1}^{D} \langle \langle h_{ab}(x,y), w_b(y) \rangle_{\mathcal{H}_b}, v_a(x) \rangle_{\mathcal{H}_a} \quad (8)$$

$$h_{ab}(x,y) = \mathbb{E}_{z\sim q}[-k_a(z,x)k_b(z,y)\partial_{ab} \log p(z) + \partial_{z_a} k_b(z,y)\partial_{z_b} k_a(z,x)] \quad (9)$$

For the proof of this theorem, see Appendix B.

4.2 DECOUPLED NEWTON SYSTEMS

Following SVN [Detommaso et al., 2018], we employ a block-diagonal approximation of the Hessian defined by

[1] The inner products here are between the functions h_{ab}, w_b, v_a in Hilbert spaces. x and y are free variables only included to show which functions share which inputs.

Theorem 1 and solve a decoupled system for the approximate Newton update w_i for each sample x_i

$$\hat{H}(x_i, x_i)w_i = -\nabla \hat{J}[\mathbf{0}](x_i) \quad \text{for } i \in 1, ..., n \tag{10}$$

where the entries of the Hessian $\hat{H}(x_i, x_i)$ are defined by the second variation coefficients $(\hat{H}(x_i, x_i))_{ab} = h_{ab}(x_i, x_i)$ from Eq. 9.

5 TRUST-REGION METHODS

Our second contribution is implementing two trust-region methods for SVI optimization. Trust-region methods are iterative procedures for optimizing a smooth objective function $f : \mathbb{R}^M \to \mathbb{R}$ [Nocedal and Wright, 1999]. At each iteration t, these methods generate an additive update $w \in \mathbb{R}^M$ to the current estimate $x \in \mathbb{R}^M$ by minimizing a second-order approximation of the objective over a closed ball defined by some norm $||\cdot||$ and a radius Δ, called the *trust region*

$$\min_{w \in \mathbb{R}^M} f(x) + \nabla f(x)^\top w + \frac{1}{2} w^\top H w \quad \text{s.t. } ||w|| \leq \Delta \tag{11}$$

where H is the symmetric Hessian model.

Given the Newton system's (Eq. 10) block-diagonal structure, we propose utilizing the norm $||w|| \triangleq \max\{||w_i||_2\}_{i=1}^n$ to define the trust region, where w_i is the update for each individual sample x_i. The advantage of using this norm is that, in combination with the block-diagonal Hessian model, the trust-region minimization Eq. 11 is *separable* over the updates w_i for each sample x_i; consequently, these updates can be efficiently computed in parallel.

5.1 KL DIVERGENCE APPROXIMATION

Standard trust-region methods rely on objective function evaluations to adjust the trust-region radius Δ used in each iteration based on the observed change in objective value. In the specific case of SVI, we aim to minimize KL divergence. The computation of KL divergence can be split into two terms

$$KL(q||p) = \mathbb{E}_{x \sim q}\left[-\log p(x)\right] - \mathcal{H}(q)$$

where $\mathcal{H}(q)$ is the entropy of q. The first term can be easily approximated via an empirical estimate with our current sample set. The second term requires computing the entropy of q given the representative set of samples $\{x_i\}_{i=1}^n$, which is more difficult. Since SVI already requires the computation of kernel matrices, we propose utilizing the kernelized approximation of entropy from Bach [2022].

$$\mathcal{H}(q) \approx -\mathrm{tr}\left[\frac{1}{n} K \log(\frac{1}{n} K)\right] = -\sum_{i=1}^{n} \lambda_i \log(\lambda_i) \tag{12}$$

where K is the $n \times n$ kernel matrix for the sample $\{x_i\}_{i=1}^n$, such that $K_{ij} = k(x_i, x_j)$, and $\{\lambda_i\}_{i=1}^n$ are the eigenvalues of $\frac{1}{n}K$. Since this approximation requires the computation of eigenvalues, which is a computationally expensive $O(n^3)$ operation, we utilize the eigenvalues of a Nyström approximation of $\frac{1}{n}K$ in place of the full matrix. We use a non-local kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ for this approximation. The full algorithmic details of the approximation are presented in Algorithm 1. It's time complexity is $\mathcal{O}(m^3 + n)$ where m is the Nyström size.

Algorithm 1 Approx-KL

1: **Inputs:** Sample points $\{x_i\}_{i=1}^n$, reference distribution p, Nyström size m
2: Select a subset $S \subset \{x_i\}_{i=1}^n$ of size m uniformly at random without replacement
3: Compute the kernel matrix K using kernel function k on the subset S
4: $U, \{\lambda_i\}_{i=1}^m, V = \mathrm{SVD}(\frac{1}{n}K)$
5: $H = \sum_{i=1}^m \lambda_i \log(\lambda_i)$
6: $P = \frac{1}{n}\sum_{i=1}^n \log p(x_i)$
7: **Return** $-P + H$

Our first trust-region algorithm, TR-SVI-KL, adjusts the shared trust-region size based on how well the local quadratic model (Eq. 11) predicts the observed change in objective value. If a step increases the objective value, it is rejected. The full algorithmic details of this trust-region method are presented in Algorithm 2. The per-iteration time complexity of this method is $\mathcal{O}(n^2D^2 + m^3)$ where m is the Nyström size used for Approx-KL.

5.2 GRADIENT-BASED TRUST-REGION

Even with the Nyström approximation, the KL divergence approximation in Algorithm 1 is still relatively expensive to compute. Moreover, the approximation error can negatively impact the efficacy of the trust-region adjustment. Therefore, in this subsection, we describe an alternative trust-region method that avoids the need to evaluate the objective at all, by taking advantage of the recently developed AdaTrust method [Grapiglia and Stella, 2022].

The motivating idea behind a gradient-based trust-region is that a converging optimization should contain a subset of iterations in which the magnitude of the gradient is consistently decreasing. Our gradient-based trust-region method stores the lowest observed gradient magnitude value and compares the gradient magnitude at each new iterate against it. The trust-region is expanded if the gradient magnitude at the current iterate is less than the lowest seen so far and constricted otherwise. The algorithmic details of this trust-region method are presented in Algorithm 3. The per-iteration time complexity of this method is the same as SVN, $\mathcal{O}(n^2D^2)$.

Algorithm 2 TR-SVI-KL

Inputs: Initial points $\{x_i\}_{i=1}^n$, reference distribution p, initial trust-region Δ
for each iteration t **do**
  for each sample x_i **do**
    Compute w_i by solving system in Eq. 10 using CG-Steihaug [Nocedal and Wright, 1999] with trust-region Δ
  end for
  $m = \sum_{i=1}^n \frac{1}{2} w_i^\top \hat{H}(x_i, x_i) w_i + \nabla \hat{J}[\mathbf{0}](x_i)^\top w_i$
  $u = \text{Approx-KL}(\{x_i + w_i\}_{i=1}^n, p, \lfloor n/10 \rfloor)$
  $o = \text{Approx-KL}(\{x_i\}_{i=1}^n, p, \lfloor n/10 \rfloor)$
  $\rho = \frac{u-o}{m}$
  $\Delta = \begin{cases} \Delta/2 & \text{if } \rho < .0001 \\ 1.5\Delta & \text{if } \rho > .7 \\ \Delta & \text{otherwise} \end{cases}$
  $\{x_i\}_{i=1}^n = \begin{cases} \{x_i\}_{i=1}^n & \text{if } \rho < 0 \\ \{x_i + w_i\}_{i=1}^n & \text{otherwise} \end{cases}$
end for

Algorithm 3 TR-SVI-AT

Inputs: Initial points $\{x_i\}_{i=1}^n$, reference distribution p
$b_{min} = .1$
$b, w, b_{max}, g = \sqrt{\sum_{i=1}^n ||\nabla \hat{J}[\mathbf{0}](x_i)||^2}$
for each iteration t **do**
  for each sample point x_i **do**
    Compute w_i by solving system in Eq. 10 using CG-Steihaug [Nocedal and Wright, 1999] with trust-region $\frac{g}{b}$
  end for
  $\{x_i\}_{i=1}^n = \{x_i + w_i\}_{i=1}^n$
  $g = \sqrt{\sum_{i=1}^n ||\nabla \hat{J}[\mathbf{0}](x_i)||^2}$
  $b, w = \begin{cases} \max(b_{min}, .9b), g & \text{if } g < .999w \\ \min(b_{max}, b + \frac{g^2}{b}), w & \text{otherwise} \end{cases}$
end for

6 RESULTS

6.1 EXPERIMENTAL SET-UP

In this section, we experimentally evaluate our trust-region SVI algorithms. As baselines, we compare against prior work on SVI, namely MP-SVGD [Zhuo et al., 2018] and SVN [Detommaso et al., 2018], which also serve as ablations of our method. Motivated by the results in Appendix F, we utilize variants with more advanced step control rules to make these baselines as strong as possible. The first of these modified methods is MP-SVGD-DSS, which utilizes a decaying step size. The second is MP-SVGD-AG which utilizes the off-the-shelf adaptive optimizer AdaGrad [Duchi et al., 2011] for step control. The third is SVN-CTR which utilizes the CG-Steihaug method [Nocedal and Wright, 1999] for solving the systems in Eq. 7 with a constant trust-region size.

The hyperparameters for the trust-region methods were set based on performance on small toy problems during early algorithm development and then used for every experiment without alteration. In general, the convergence of trust-region methods is not sensitive to exact hyperparameter settings [Nocedal and Wright, 1999]. The hyperparameters for the SVI baselines were fit to each specific problem via grid search to maximize performance, which introduces extra information that our methods did not receive. Note that this makes the following comparisons somewhat unfairly biased against our methods. All SVI methods, both ours and the baselines, utilize an RBF kernel with a lengthscale set manually based on performance. Exact hyperparameter settings for each method and problem are shown in Appendix E. We also compare against VIPS40 [Arenz et al., 2018], a state-of-the-art, Gaussian mixture model-based variational inference method. All experiments were run on a desktop with an Intel Core i7-13700K, 32 GB RAM, and a NVIDIA RTX 4080.

6.2 BAYES NET EXPERIMENT

Our first set of experiments is designed to test the different variational inference methods' ability to scale to distributions that are high-dimensional and poorly conditioned in a controlled environment where we can easily recover ground truth samples from the reference distribtuion. To this end, we evaluate performance on recovering the joint density of a Bayes net [Koller and Friedman, 2009]. This synthetic problem gives us a high degree of control over the parameters of the distribution and the generative nature of Bayes nets enables us to easily recover a ground truth sample via ancestral sampling. This problem also has known conditional independence structure that our methods and graphical baselines can exploit.

The nodes of the Bayes net are organized into layers with nodes in each layer conditioned only on nodes from the previous layer. The conditional/marginal distributions encoded by each node are either a Gaussian or Gaussian mixture. Note that this makes the resulting joint of the Bayes net a Gaussian mixture. The exact parameters of each node are generated randomly according to the generative process described in Appendix D. The generative parameters for the Gaussian mixture nodes are selected to encourage distinct modes. The nodes are also generated to have vastly different variances to induce poor-conditioning. We test on two Bayes nets examples, one 30-dimensional and one 80-dimensional. A ground truth sample for each example, containing 6 million points, was generated via ancestral sampling. Each variational inference method was used to produce a sample

Table 1: Maximum Mean Discrepancies against Ground Truth Sample

Model	12-Dimensional SNLP	30-Dimensional BN	80-Dimensional BN
MP-SVGD-DSS	0.05276 ± 0.03560	0.1492 ± 0.00618	0.2251 ± 0.00577
MP-SVGD-AG	$0.05091 \pm .01328$	0.2130 ± 0.02086	0.2004 ± 0.00975
SVN-CTR	0.05067 ± 0.00937	0.1887 ± 0.00877	0.2707 ± 0.00276
VIPS40	0.1981 ± 0.12383	$\mathbf{0.00185 \pm 0.00095}$	0.2565 ± 0.03055
TR-SVI-KL	0.04800 ± 0.00979	0.01496 ± 0.00741	0.08634 ± 0.02348
TR-SVI-AT	$\mathbf{0.03530 \pm 0.01366}$	0.009674 ± 0.00623	$\mathbf{0.07646 \pm 0.02051}$

with 200 points.

We evaluate the quality of each method's sample by comparing against the ground truth sample using the maximum mean discrepancy (MMD) metric [Gretton et al., 2012]. MMD was chosen because it is designed to find the test statistic that reveals the greatest discrepancy between two distributions. MMD is also kernel-based which makes it a somewhat natural choice to evaluate SVI methods. The MMD of two samples $X = \{x_i\}_{i=1}^n$ and $Y = \{y_i\}_{i=1}^m$ is computed as

$$\text{MMD}(X,Y) = \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i,x_j) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k(x_i,y_j) + \frac{1}{m^2}\sum_{i,j=1}^{m} k(y_i,y_j) \quad (13)$$

For our tests, the kernel k is the RBF kernel whose length-scale is set using the median heuristic [Garreau et al., 2017] on the ground truth sample. The performance for each variational inference method on this metric is shown in Table 1. All numerical results are averaged over 5 runs with randomly initialized positions for the SVI particle sets. The standard deviation of the MMD statistic over these runs is reported in the error bars.

Our methods outperformed the SVI baselines on both Bayes net problems, suggesting better scaling to high dimensions and poor conditioning. VIPS40 did outperform our method on the lower-dimensional Bayes net problem. That said, since the joint distribution of this Bayes net is a Gaussian mixture, the Gaussian mixture-based approximation employed by VIPS40 is the information-theoretically optimal choice in this specific case. The difference in performance between our methods and VIPS40 on this problem represents the cost of flexibility. Although our methods can more accurately capture a wider variety of distribution shapes, they cannot achieve gold-standard performance of parametric models in their ideal use cases. Furthermore, our method scales more effectively than VIPS40 in higher dimensions, performing significantly better on the 80-dimensional example. We attribute the pronounced decrease in VIPS40's performance to the fact that VIPS40 does not take advantage of the conditional independences present in these Bayes net models, the exploitation of which is well-known to be critical for achieving efficient inference in high dimensions.

6.3 LOW-DIMENSIONAL SNLP WITH GROUND TRUTH

Our next set of experiments is designed to highlight the flexibility of our methods by investigating both the convergence behavior and posterior approximation quality of the different variational inference methods on a real-world problem with complex posterior shapes. To this end, we evaluate our methods and the baselines on a small example of the sensor network localization problem (SNLP) [Biswas et al., 2006]. This problem tends to have posteriors with multimodal and annular shapes and is therefore hard for parametric models to accurately approximate. Its low dimensionality enables us to recover a ground truth sample for quantitative analysis.

The goal of the SNLP is to recover the positions of a set of sensors $S = \{s_1, \ldots, s_n\} \subset \mathbb{R}^D$ from a given set of noisy pairwise distance measurements between them. This problem can be modeled as a graph $G = (V, E)$ in which the sensors are represented by the vertices V and the available measurements $\tilde{d}_{ij}$ are represented by the edges E between them. A pair of sensors s_i and s_j can only generate a range measurement when the distance $||s_i - s_j||$ between them is less than some maximum effective sensing radius r, and any such measurement is generated according to:

$$\tilde{d}_{ij} = ||s_i - s_j|| + \epsilon_{ij} \qquad \epsilon_{ij} \sim N(0, \sigma^2)$$

Finally, we assume that there is a subset $A \subset S$ of the sensors (called the *anchors*) whose positions are known exactly *a priori*.

For these experiments, we use an SNLP instance with 6 estimated nodes and 4 anchors placed on a 6×6 unit square in $\mathbb{R}^2$. The maximum range is $r = 3$ and the measurements are noiseless but believed to be corrupted by noise $\epsilon_{ij} \sim \mathcal{N}(0, .01)$. The problem is visualized in graphical form in Appendix C. A ground truth sample containing about 1.1 million points was recovered using a standard nested sampling library, dynesty [Speagle, 2020]. Each variational inference method was set to produce a sample of

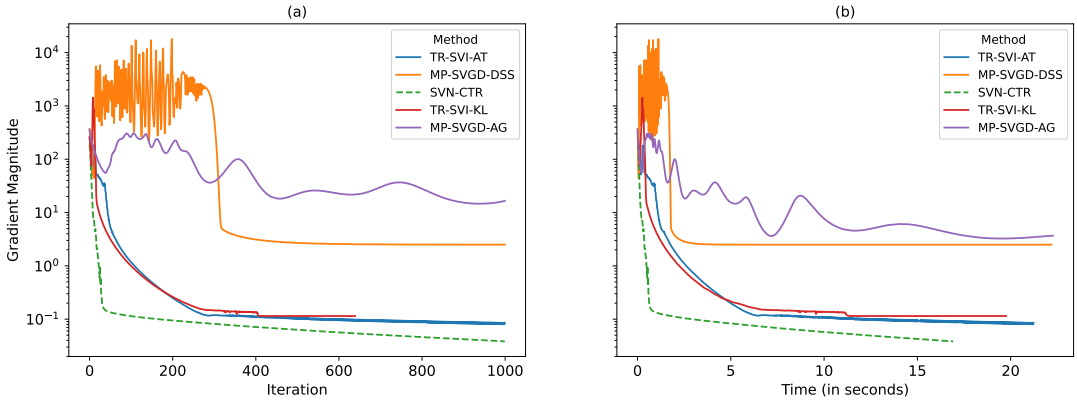

Figure 1: The convergence rate as a function of iteration number (a) and compute time (b) of each SVI method on the small SNLP instance. All second-order methods show fast, smooth convergence both MP-SVGD variants oscillate until their step size decays enough to enable convergence. Note that, unlike the other methods, SVN-CTR does not use local kernels to compute the gradient (see Eqs. 3 and 5). Since the estimated gradients depend upon the choice of kernel, SVN-CTR's gradient magnitude values are not directly comparable.

200 points. Although this problem is relatively small, it is sufficient to reveal significant differences in performance between the different variational inference methods.

6.3.1 Convergence Results

Our first experiment in this set focuses on analyzing the convergence behavior of the different SVI methods with different step control approaches. Methods like SVN and MP-SVGD that lack adaptive step control rely on *a priori* user-specified step control rules or off-the-shelf adaptive optimizers. User-specified step control rules and some popular off-the-shelf adaptive optimizers, like Adam [Reddi et al., 2018], lack the strong convergence guarantees of trust-region methods [Nocedal and Wright, 1999]. AdaGrad [Duchi et al., 2011], another popular off-the-shelf optimizer, has convergence guarantees but monotonically decreases the step size, making it less adaptive and often slower to converge than trust-region methods. These experiments demonstrate the performance gap in convergence behavior between trust-region methods and these other step control methods.

Figure 1a depicts the convergence for our methods and the modified baselines as a function of iteration number. All the second-order methods converge quickly and smoothly. Both MP-SVGD variants oscillate until the step size decays enough to allow convergence. TR-SVI-KL shows fewer iterations than the other methods because of its step rejection mechanism, which the other methods lack.

Figure 1b depicts the convergence of the methods as a function of time. Although the per-iteration time complexity of second-order methods is greater than that of first-order methods, they require fewer iterations, and thus less compute time overall, to achieve a small gradient magnitude.

6.3.2 Numerical Performance of Generated Samples

Next, we evaluate the accuracy of the generated sample sets as approximations of the target posterior distribution, once again using the MMD metric (Eq. 13) against the dynesty reference sample. The results of this experiment are presented in Table 1.

Our trust-region methods achieved the lowest average MMD scores with the reference sample. VIPS40 performed the worst on this test, potentially due to the difficulty of approximating the annular shapes of the SNLP posterior with a Gaussian mixture. The samples produced by the different methods are assessed qualitatively in Appendix G.

6.4 QUALITATIVE ANALYSIS OF HIGH-DIMENSIONAL SNLP

Our last set of experiments is intended to test how the different methods scale to a high-dimensional problem with complex shapes. To this end, we apply it to a larger 50 sensor, 12 anchor SNLP instance over a 20×20 unit square in $\mathbb{R}^2$. For this problem, the maximum sensor range $r = 3$ and the measurements received by the sensors are perturbed with noise $\epsilon_{ij} \sim \mathcal{N}(0, .01)$ whose mean and variance are known values. The results of this experiment are only assessed qualitatively since recovering a high-quality reference sample using traditional methods (e.g. dynesty [Speagle, 2020]) is intractable. For comparison, MP-SVGD-DSS, SVN-CTR, and VIPS40 were also run on this test. The same set of noisy measurements were used for all methods to ensure a fair comparison.

Figure 2 shows KDE plots generated from the samples produced by each of four methods for three selected sensors and

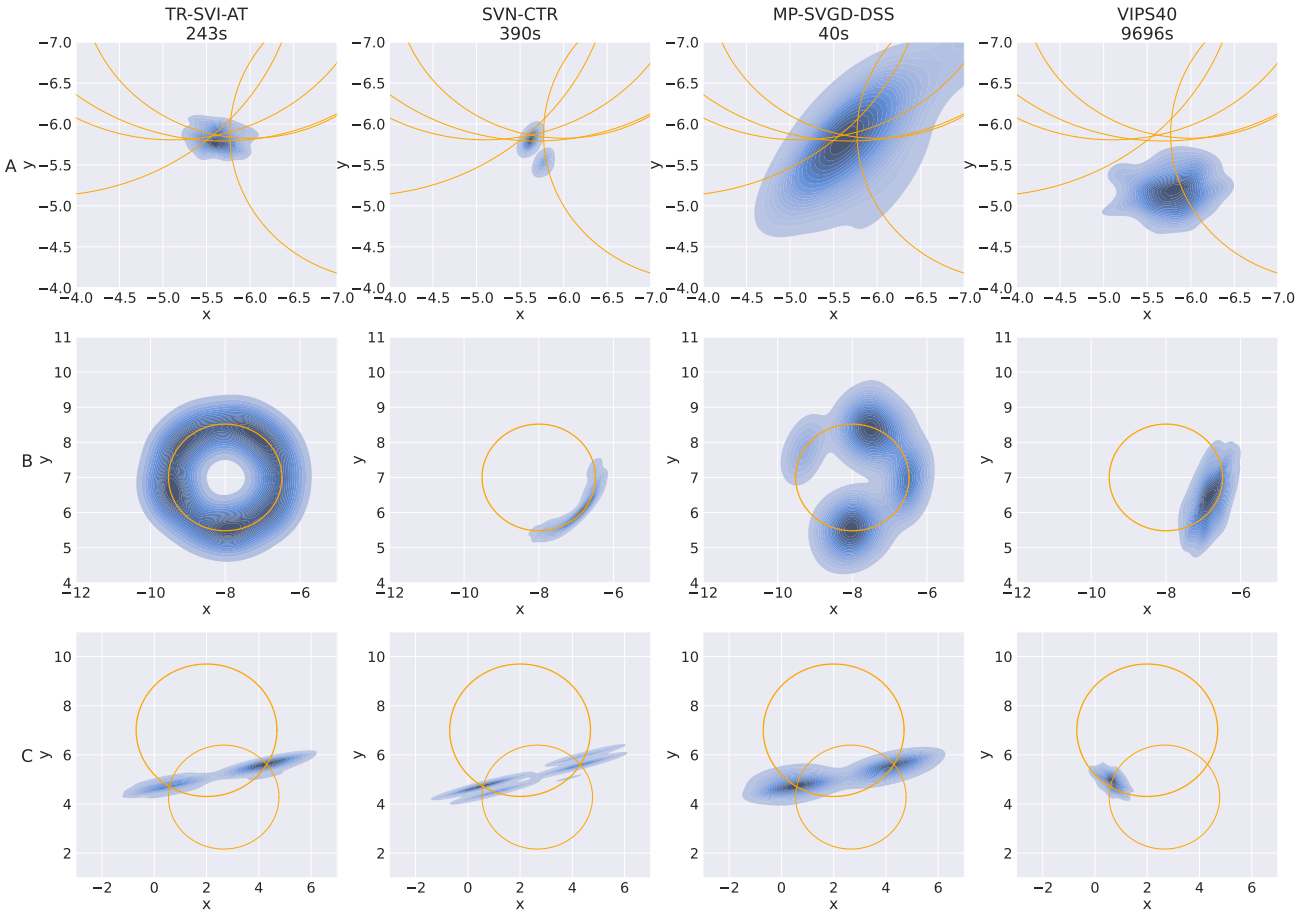

Figure 2: Kernel density estimation (KDE) plots of the final samples produced by various variational inference methods on a high-dimensional, noisy SNLP problem. From each sample, the marginal samples corresponding to the location of a selected sensor are extracted and visualized as a KDE plot. Since ground truth was not recoverable, we also visualize the measurements received by each selected sensor to enable qualitative analysis. These measurements are displayed as orange circles with a radius equal to the range measurement centered on the true position of the sending node. The time to generate the sample (in seconds) is displayed under its name.

the time required to generate each approximation. Figure 2 also displays the incoming measurements for each sensor as circles to enable easier qualitative analysis.

Sensor A received multiple measurements, so a unimodal distribution is expected, with some variation due to sensor noise. All SVI methods recovered a unimodal distribution centered correctly on the intersection of the various measurements. VIPS40 recovered a unimodal distribution, but it is not correctly centered. Sensor B received a single range measurement from an anchor, so its posterior should be annular. Of the four methods, only TR-SVI-AT produced a sample with a balanced annular shape. Sensor C received two range measurements, resulting in a bimodal distribution. All SVI methods captured this bimodal distribution but VIPS40 only captured one of the two modes.

Overall, TR-SVI-AT appears to capture intricate details in high dimensions significantly more accurately than previous variational inference methods. Notably, VIPS40 took significantly (>20x) longer than the SVI methods and produced a visibly worse approximation.

7 CONCLUSION

We introduce a SVI method that leverages known conditional independence structure, second-order information, and adaptive step control to ensure good convergence on high-dimensional, non-convex, and ill-conditioned objectives. Our method demonstrated faster and more reliable convergence than existing SVI methods. The approximations produced by our method were more accurate than those of existing SVI methods and a state-of-the-art parametric variational inference method.

Acknowledgements

Liam Pavlovic was supported by the National Science Foundation Graduate Research Fellowship Program.

References

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50, 2003.

Oleg Arenz, Gerhard Neumann, and Mingjun Zhong. Efficient gradient-free variational inference using policy search. In *International conference on machine learning*. PMLR, 2018.

Francis Bach. Information theory with kernel methods. *IEEE Transactions on Information Theory*, 69(2), 2022.

Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.

Pratik Biswas, T-C Liang, K-C Toh, Yinyu Ye, and T-C Wang. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE transactions on automation science and engineering*, 3(4), 2006.

Peng Chen and Omar Ghattas. Projected Stein variational gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Peng Chen, Keyi Wu, Joshua Chen, Tom O'Leary-Roseberry, and Omar Ghattas. Projected Stein variational Newton: A fast and scalable Bayesian inference method in high dimensions. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Gianluca Detommaso, Tiangang Cui, Youssef Marzouk, Alessio Spantini, and Robert Scheichl. A Stein variational Newton method. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

Damien Garreau, Wittawat Jitkrittum, and Motonobu Kanagawa. Large sample analysis of the median heuristic. *arXiv preprint arXiv:1707.07269*, 2017.

Wenbo Gong, Yingzhen Li, and José Miguel Hernández-Lobato. Sliced kernelized Stein discrepancy. In *International Conference on Learning Representations*, 2021.

Geovani N Grapiglia and Gabriel FD Stella. An adaptive trust-region method without function evaluations. *Computational Optimization and Applications*, 82(1), 2022.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1), 2012.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in neural information processing systems*, volume 29, 2016.

Xing Liu, Harrison Zhu, Jean-Francois Ton, George Wynne, and Andrew Duncan. Grassmann Stein variational gradient descent. In *International Conference on Artificial Intelligence and Statistics*, volume 151. PMLR, 2022.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations*, 2018.

Joshua S Speagle. DYNESTY: a dynamic nested sampling package for estimating Bayesian posteriors and evidences. *Monthly Notices of the Royal Astronomical Society*, 493 (3), 2020.

Dilin Wang, Zhe Zeng, and Qiang Liu. Stein variational message passing for continuous graphical models. In *International Conference on Machine Learning*. PMLR, 2018.

Yating Wang, Wei Deng, and Guang Lin. An adaptive Hessian approximated stochastic gradient MCMC method. *Journal of Computational Physics*, 432:110150, 2021.

Yifei Wang and Wuchen Li. Information Newton's flow: second-order optimization method in probability space. *arXiv preprint arXiv:2001.04341*, 2020.

Jingwei Zhuo, Chang Liu, Jiaxin Shi, Jun Zhu, Ning Chen, and Bo Zhang. Message passing Stein variational gradient descent. In *International Conference on Machine Learning*. PMLR, 2018.

Appendix

Liam Pavlovic[1] **David M. Rosen**[1]

[1]Northeastern University, Boston, Massachussets, USA

A PROOF OF EQUATION 5

From the proof of Theorem 3.3 in Liu and Wang [2016], we have

$$\tau\langle\nabla J[S], V\rangle + O(\tau^2) = J[S + \tau V] - J[S]$$

and, at $S = \mathbf{0}$,

$$J[\mathbf{0} + \tau V] - J[\mathbf{0}] = -\Delta_1 - \Delta_2$$

where

$$\begin{aligned}\Delta_1 &= \mathbb{E}_{z\sim q}[\log p(z + \tau V(z))] - \mathbb{E}_{z\sim q}[\log p(z)] \\ \Delta_2 &= \mathbb{E}_{z\sim q}[\log\det(I + \tau\nabla_z V(z))] - \mathbb{E}_{z\sim q}[\log\det(I)]\end{aligned}$$

For $V \in \mathcal{H}_1 \times ... \times \mathcal{H}_D$ the terms above can be computed as

$$\begin{aligned}\Delta_1 &= \mathbb{E}_{z\sim q}[\log p(z + \tau V(z))] - \mathbb{E}_{z\sim q}[\log p(z)] \\ &= \tau\mathbb{E}_{z\sim q}[\nabla_z \log p(z) \cdot V(z)] + O(\tau^2) \\ &= \tau\sum_{a=1}^{D}\mathbb{E}_{z\sim q}[\nabla_{z_a}\log p(z)v_a(z)] + O(\tau^2) \\ &= \tau\sum_{a=1}^{D}\langle\mathbb{E}_{z\sim q}[\nabla_{z_a}\log p(z)k_a(z,\cdot)], v_a(\cdot)\rangle_{\mathcal{H}_a} + O(\tau^2)\end{aligned}$$

and

$$\begin{aligned}\Delta_2 &= \mathbb{E}_{z\sim q}[\log\det(I + \tau\nabla_z V(z))] - \mathbb{E}_{z\sim q}[\log\det(I)] \\ &= \tau\mathbb{E}_{z\sim q}[\text{trace}(I^{-1}\cdot\nabla_z V(z))] + O(\tau^2) \\ &= \tau\mathbb{E}_{z\sim q}[\text{trace}(\nabla_z V(z))] + O(\tau^2) \\ &= \tau\sum_{a=1}^{D}\mathbb{E}_{z\sim q}[\nabla_{z_a} v_a(z)] + O(\tau^2) \\ &= \tau\sum_{a=1}^{D}\langle\mathbb{E}_{z\sim q}[\nabla_{z_a} k_a(z,\cdot)], v_a(\cdot)\rangle + O(\tau^2)\end{aligned}$$

Thus,

$$\langle\nabla J[\mathbf{0}], V\rangle = \sum_{a=1}^{D}\langle -\mathbb{E}_{z\sim q}[k_a(z,\cdot)\nabla_{z_a}\log p(z) + \nabla_{z_a}k_a(z,\cdot)], v_a\rangle_{\mathcal{H}_a}$$

and

$$(\nabla J[\mathbf{0}](\cdot))_a = -\mathbb{E}_{z\sim q}[k_a(z,\cdot)\nabla_{z_a}\log p(z) + \nabla_{z_a}k_a(z,\cdot)]$$

B PROOF OF THEOREM 1

From the proof of Theroem 1 in Detommaso et al. [2018], we know that the second variation of the SVI objective along a pair of directions $V, W \in \mathcal{H}_1 \times ... \times \mathcal{H}_D$ equals

$$D^2 J[\mathbf{0}](V, W) = -\mathbb{E}_{z\sim q}\left[W(z)^\top \nabla_z^2 \log p(z) V(z) - \text{trace}(\nabla_z W(z) \nabla_z V(z))\right] \tag{14}$$

By the reproducing properties of $\mathcal{H}_1 \times ... \times \mathcal{H}_D$, namely

$$v_a(z) = \langle k_a(z, \cdot), v_a(\cdot)\rangle_{\mathcal{H}_a} \qquad w_a(z) = \langle k_a(z, \cdot), w_a(\cdot)\rangle_{\mathcal{H}_a}$$

and

$$\partial_{z_b} v_a(z) = \langle \partial_{z_b} k_a(z, \cdot), v_a(\cdot)\rangle_{\mathcal{H}_a} \qquad \partial_{z_b} w_a(z) = \langle \partial_{z_b} k_a(z, \cdot), w_a(\cdot)\rangle_{\mathcal{H}_a}$$

we get

$$\mathbb{E}_{z\sim q}\left[W(z)^\top \nabla_z^2 \log p(z) V(z)\right] = \sum_{a=1}^{D}\sum_{b=1}^{D} \langle\langle \mathbb{E}_{z\sim q}\left[-k_a(z, x) k_b(z, y) \partial_{ab}^2 \log p(z)\right], w_b(y)\rangle_{\mathcal{H}_b}, v_a(x)\rangle_{\mathcal{H}_a}$$

and

$$\mathbb{E}_{z\sim q}\left[\text{trace}(\nabla_z W(z) \nabla_z V(z))\right] = \sum_{a=1}^{D}\sum_{b=1}^{D} \langle\langle \mathbb{E}_{z\sim q}\left[\partial_{z_a} k_b(z, y) \partial_{z_b} k_a(z, x)\right], w_b(y)\rangle_{\mathcal{H}_b}, v_a(x)\rangle_{\mathcal{H}_a}$$

Plugging these in yields the final expression for the second variation

$$\sum_{a=1}^{D}\sum_{b=1}^{D} \langle\langle h_{ab}(x, y), w_b(y)\rangle_{\mathcal{H}_b}, v_a(x)\rangle_{\mathcal{H}_a} \tag{15}$$

where

$$h_{ab}(x, y) = \mathbb{E}_{z\sim q}\left[-k_a(z, x) k_b(z, y) \partial_{ab} \log p(z) + \partial_{z_a} k_b(z, y) \partial_{z_b} k_a(z, x)\right] \tag{16}$$

C SNLP PROBLEM VISUALIZATIONS

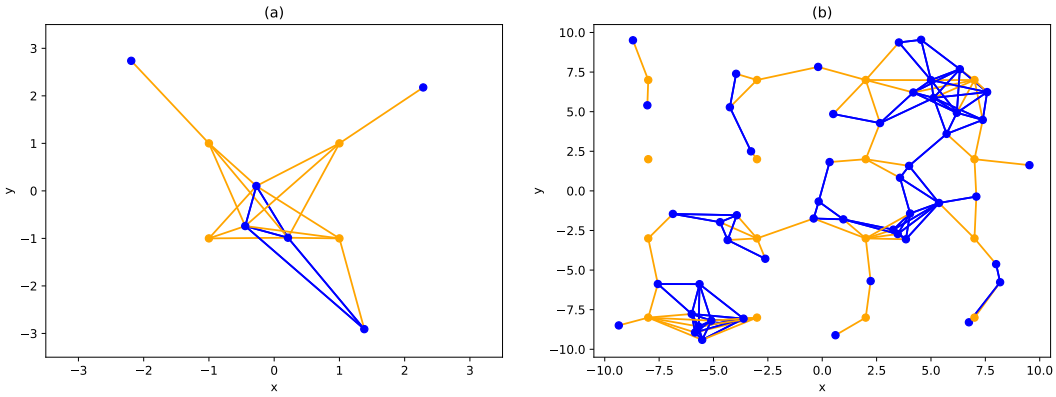

Figure 3: Graph representations of the sensor network localization problems used for evaluation with the small example on the left and the large example on the right. Estimated nodes are depicted in blue and anchors in orange. The edges represent shared range measurements between pairs of nodes. Blue edges correspond to measurements shared between two estimated nodes and orange edges correspond to measurements from an anchor.

D BAYES NET GENERATION DETAILS

The nodes of these Bayes Nets are organized into layers. The 30-dimensional has 3 layers with 10 nodes each. The 80-dimensional has 4 layers with 20 nodes each. A node x_j from the first layer has a marginally Gaussian distribution

$p(x_j) = \mathcal{N}(\mu, \sigma^2)$. A node x_j in any subsequent layer is conditioned on some random $[1, M]$-size subset C_j of the nodes from the previous layer with which it shares connections in the network. The conditional distribution of such a node is either a Gaussian or Gaussian mixture of the form

$$p(x_j|C_j) = \mathcal{N}(\sum_{x_k \in C_j} \alpha_k x_k, \sigma^2) \quad \text{or} \quad p(x_j|C_j) = \sum_{l=1}^{2} \omega_l \mathcal{N}(\sum_{x_k \in C_j} \alpha_k^l x_k, \sigma^2) \tag{17}$$

where $\{\alpha_k^{(l)}\}$ is a set of weights and $\{\omega_l\}$ are the GMM component weights. The 30-dimensional problem has a total of 6 GMM nodes and the 80-dimensional has 20. Nodes from all layers but the first were uniformly selected at random to be a GMM node. The specific random generative procedure for values of each parameter are

- μ is selected uniformly from $[0, 2]$ for 30-dimensional $[0, 4]$ for 80-dimensional
- All weights $\{\alpha_k^{(l)}\}$ are selected independently and uniformly from $[-1, 1]$
- The first GMM weight ω_1 is selected uniformly from $[.4, .6]$, the second ω_2 completes the sum to 1.
- All variances σ^2 were sampled uniformly over orders of magnitude $[10^{-3}, 10^0]$ to induce poor-conditioning
- Maximum number of connections M is 3 for 30-dimensional and 4 for 80-dimensional

E HYPERPARAMETER DETAILS

Table 2: Hyperparameter Settings

Model	12-Dim SNLP	100-dim SNLP	30-Dim BN	80-Dim BN
MP-SVGD-DSS (initial step, step decay)	0.1, 0.99	0.1, 0.99	0.01, 0.999	0.01, 0.99
MP-SVGD-AG (intial step)	0.5	N/A	0.05	0.05
SVN-CTR (trust region size)	1	0.1	0.1	0.1
Kernel (lengthscale)	1	3	10	60

The hyperparameters used for each SVI baseline, as well as the kernel hyperparameters utilized by all SVI methods, are shown in Table 2. VIPS40 utilizes the default configuration provided by the source code [Arenz et al., 2018]. TR-SVI-AT and TR-SVI-KL utilize the parameter settings listed in their respective algorithm blocks (Algorithms 2, 3) for all experiments.

F ADDITIONAL CONVERGENCE RESULTS

This additional set of tests is designed to demonstrate the necessity of adaptive step control for ensuring convergence on non-convex objectives. To do this, we analyze the convergence of MP-SVGD [Zhuo et al., 2018] and SVN [Detommaso et al., 2018] on the small SNLP example with static step sizes. Figure 4 depicts the convergence rates of MP-SVGD and SVN with a variety of static step sizes, none of which produce good results. For MP-SVGD, the step size is either too large for a portion of the optimization, causing the method to oscillate over the objective, or too small, resulting in slow convergence. For SVN, all the step sizes result in overly large initial steps from which the method subsequently struggles to recover. The poor performance of these methods in this experiment motivated the introduction of the improved MP-SVGD-DSS, SVN-CTR and MP-SVGD-AG baselines.

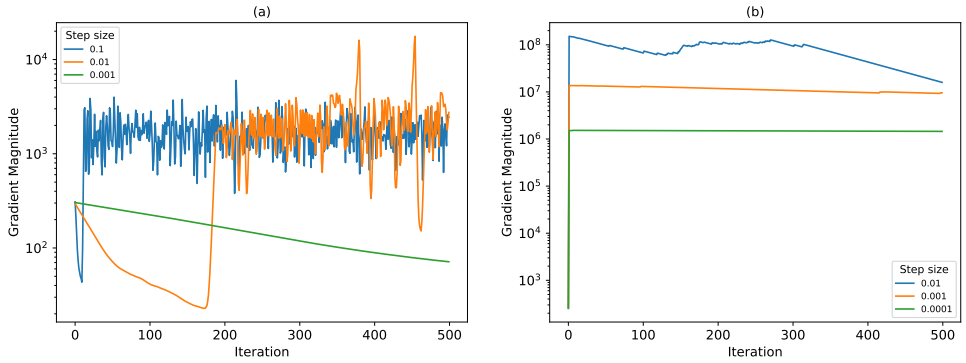

Figure 4: The convergence rates of MP-SVGD(a) and SVN(b) on the small SNLP instance with a variety of static step sizes. None produce good results.

G QUALITATIVE ANALYSIS OF LOW-DIMENSIONAL SNLP

This additional set of tests is designed to demonstrate how differences in MMD performance on the small SNLP instance, as reported in Table 1 translate to perceptible differences in the quality of the different sample approximations. Figure 5 shows kernel density estimate (KDE) plots generated from samples produced by MP-SVGD-DSS, SVN-CTR, TR-SVI-AT, VIPS40, and dynesty, as well as the time required to generate the samples. Plots of the marginal posterior estimates for the locations of three selected sensors are shown. These sensors were selected because they represent a variety of posterior shapes.

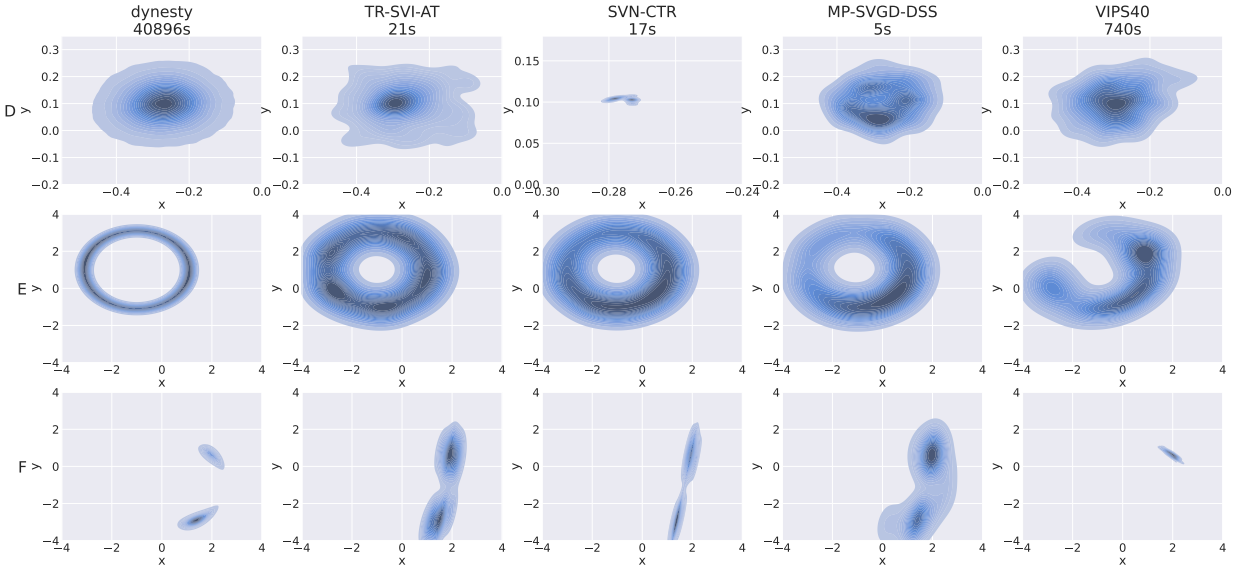

Figure 5: Kernel density estimation (KDE) plots of the final samples produced by various SVI methods and the dynesty reference on a low-dimensional SNLP problem. From each sample, the marginal samples corresponding to the location of the selected sensor are extracted and visualized as a KDE plot. The KDE plots of the different methods are displayed on the same scale with the exception of SVN-CTR's plot for sensor A, which required a scale an order of magnitude smaller to be visible. The time required to generate each sample is displayed under its name.

Sensor D received multiple measurements from other nodes, resulting in a dense unimodal distribution. VIPS40, TR-SVI-AT, and MP-SVGD-DSS capture this sensor's posterior well. SVN-CTR, on the other hand, significantly underestimates the variance of the posterior, requiring a different axis scale than the reference distribution to be visible.

Sensor E received single anchor measurement resulting in a annular posterior. All three SVI methods produced a sample with a annular shape, but the samples of MP-SVGD-DSS and SVN-CTR display a bias towards the bottom right of the

annulus not present in TR-SVI-AT's sample. VIPS40 recovers only a partial arc that is misshapen. All 4 methods produce more diffuse distributions than the reference.

Sensor F received a few range measurements but not as many as Sensor D, resulting in a bimodal distribution. All 3 SVI methods capture the bimodality of the distribution, but they also assign more probability to the lower probability mode than the reference distribution. VIPS40 does not capture the bimodality of this distribution, only capturing the lower probability mode.