Sparse Inducing Points in Deep Gaussian Processes: Enhancing Modeling with Denoising Diffusion Variational Inference

Jian Xu[1] Delu Zeng[1] John Paisley[2]

[1]South China University of Technology, Guangzhou, China
[2]Columbia University, New York, USA

ICML 2024

Background

- **Deep Gaussian Processes (DGPs)**:
  - Extend Gaussian Processes to multiple layers to capture hierarchical structures (Damianou & Lawrence, 2013) [1].
  - Crucial aspect: Selection of inducing variables.
  - Reduce computational burden.
- **Variational Inference Methods**:
  - Aim to approximate the true posterior distribution by minimizing KL divergence.
  - Traditional methods include Mean-field Gaussian Variational Inference (DSVI) (Salimbeni & Deisenroth, 2017) [3] and Implicit Posterior Variational Inference (IPVI) (Yu et al., 2019) [6].
- **Limitations of Traditional Methods**:
  - **DSVI**: Simplifying assumptions fail to capture complex dependencies.
  - **IPVI**: Adversarial learning leads to instability and significant bias.

Proposed Method: DDVI

- Inspired by denoising diffusion models.
- Utilizes denoising diffusion SDE and principles similar to score matching.
- Incorporates the mathematical theory of SDEs and the bridge process trick.
- Derives a variational lower bound for the marginal likelihood function.

Contributions

- Novel parameterization approach for the posterior distribution of inducing points in DGPs.
- Guarantees model efficiency and facilitates optimization and training.
- Empirically demonstrate effectiveness through extensive experiments and comparisons with baseline methods.

Gaussian Process

Consider a random function $f : \mathbb{R}^D \to \mathbb{R}$ that maps N training inputs $\mathbf{X} \triangleq \{\mathbf{x}_n\}_{n=1}^N$ to a set of noisy observed outputs $\mathbf{y} \triangleq \{y_n\}_{n=1}^N$. A zero mean Gaussian Process (GP) prior is assumed for the function, $f \sim \mathcal{GP}(0, k)$, where k denotes the covariance kernel function $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$.

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K_{XX}}) \tag{1}$$

Sparse Gaussian Processes

Sparse methods (M. K. Titsias, 2009) [5] introduce inducing points $\mathbf{Z} = \{\mathbf{z}_m\}_{m=1}^{M}$ from the input space, along with corresponding inducing variables: $\mathbf{u} = \{f(\mathbf{z}_m)\}_{m=1}^{M}$. These methods reduce the computational complexity to $\mathcal{O}(NM^2)$.

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{f}|\mathbf{K_{xz}}\mathbf{K_{zz}^{-1}}\mathbf{u}, \mathbf{K_{xx}} - \mathbf{K_{xz}}\mathbf{K_{zz}^{-1}}\mathbf{K_{zx}}) \tag{2}$$

and $p(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K_{zz}})$ is the prior over the outputs of the inducing points.

Deep Gaussian Processes

Deep Gaussian Processes are hierarchical models composed by stacking multiple-output Sparse Gaussian Processes (SGPs). Each layer consists of independent random functions, where the output of one layer serves as input to the next. Specifically, the output $\mathbf{F}_\ell$ of layer ℓ is defined as:

$$\mathbf{F}_\ell = \{f_{\ell,1}(\mathbf{F}_{\ell-1}), \cdots, f_{\ell,D_\ell}(\mathbf{F}_{\ell-1})\}, \tag{3}$$

with $f_{\ell,d} \sim \mathcal{GP}(0, k_\ell)$ being Gaussian processes, and $\mathbf{F}_0 = \mathbf{X}$. Inducing points and variables for each layer are denoted as $\mathcal{Z}$ and $\mathcal{U}$:

$$\mathcal{U} = \{\mathbf{U}_\ell\}_{\ell=1}^{L}, \quad \mathbf{U}_\ell = \{f_{\ell,1}(\mathbf{Z}_\ell), \cdots, f_{\ell,D_\ell}(\mathbf{Z}_\ell)\}. \tag{4}$$

The joint density model is given by:

$$p(\mathbf{y}, \mathbf{F}, \mathbf{U}) = p(\mathbf{y}|\mathbf{F}_L) \prod_{\ell=1}^{L} p(\mathbf{F}_\ell|\mathbf{F}_{\ell-1}, \mathbf{U}_\ell) p(\mathbf{U}). \tag{5}$$

Parameterizing Inducing Point Posteriors

Let $H = D \times M \times L$ denote the dimension of the inducing points. We aim to sample from the true posterior distribution $q(\mathbf{U})$ in $\mathbb{R}^H$, $q(\mathbf{U}) = p(\mathbf{U}|\mathbf{y})$. We start by sampling from a fixed distribution p_{fix} and then follow a Markov process in which we consider a sequential latent variable model with a joint distribution denoted as $\mathcal{Q}(\mathbf{U}_0, \ldots, \mathbf{U}_T)$, for step $t_s \in \{0, .., T-1\}$,

$$\mathbf{U}_{t_s+1} \sim \mathcal{T}(\mathbf{U}_{t_s+1} \mid \mathbf{U}_{t_s}), \quad \mathbf{U}_0 \sim p_{\text{fix}} \tag{6}$$

Time-reversal Representation of Diffusion SDE

We constrain the Markov process $\mathcal{Q}(\mathbf{U}_0, \ldots, \mathbf{U}_T)$ to be a time-reversal process of the following forward noising diffusion stochastic differential equation (SDE),

$$\mathrm{d}\overrightarrow{\mathbf{U}}_t = \mathbf{h}(t, \overrightarrow{\mathbf{U}}_t)\mathrm{d}t + g(t)\mathrm{d}B_t, \tag{7}$$

$$\overrightarrow{\mathbf{U}}_0 \sim q \quad \textbf{complicating direct sampling} \tag{8}$$

The time-reversal representation of Eq. (7) satisfies,

$$\begin{aligned} \mathrm{d}\overleftarrow{\mathbf{U}}_t = \; & g(T-t)^2 \nabla \ln\left(p_{T-t}(\overleftarrow{\mathbf{U}}_t)\right)\,\mathrm{d}t - \mathbf{h}(T-t, \overleftarrow{\mathbf{U}}_t)\,\mathrm{d}t \\ & + g(T-t)\,\mathrm{d}W_t \\ \overleftarrow{\mathbf{U}}_0 \sim p_T \approx p_{\mathrm{fix}} &= \mathcal{N}(0, \sigma^2 I) \end{aligned} \tag{9}$$

Score Matching Technique

The Score Matching Technique in diffusion-based generative modeling approximates intractable score functions to simulate the target distribution q. This is done by parameterizing the score function $s_\phi(t, \cdot)$ with neural networks and minimizing the Kullback-Leibler divergence $\mathrm{KL}(\mathcal{P}||\mathcal{P}^\phi)$. Unlike traditional score matching, which samples from p_0, here p_0 is the posterior q, complicating direct sampling. Instead, the approach minimizes $\mathrm{KL}(\mathcal{P}^\phi||\mathcal{P})$, equivalent to $\mathrm{KL}(\mathcal{Q}^\phi||\mathcal{Q})$, using samples from $\mathcal{Q}_t^\phi$. A key challenge is accurately computing $\nabla \ln \left(p_{T-t}(\overleftarrow{\mathbf{U}}_t^\phi)\right)$ due to the nonlinear drift function in the stochastic differential equation. Therefore, we need to introduce the Bridge Process Trick to compute this KL divergence.

Bridge Process Trick

KL Divergence Decomposition:

$$\mathrm{KL}(\mathcal{P}^{\phi}\|\mathcal{P}) = \mathbb{E}_{\mathcal{P}^{\phi}} \log \frac{\mathrm{d}\mathcal{P}^{\phi}}{\mathrm{d}\mathcal{P}} = \mathbb{E}_{\mathcal{P}^{\phi}} \log \frac{\mathrm{d}\mathcal{P}^{\phi}}{\mathrm{d}\mathcal{P}^{\mathrm{Bri}}} + \mathbb{E}_{\mathcal{P}^{\phi}} \log \frac{\mathrm{d}\mathcal{P}^{\mathrm{Bri}}}{\mathrm{d}\mathcal{P}} \tag{10}$$

Bridge Process Definition: The bridge process $\mathcal{P}^{\mathrm{Bri}}$ follows a specific diffusion formula:

$$\mathrm{d}\overrightarrow{\mathbf{U}}_t^{\mathrm{Bri}} = \mathbf{h}(t, \overrightarrow{\mathbf{U}}_t^{\mathrm{Bri}})\mathrm{d}t + g(t)\mathrm{d}B_t, \tag{11}$$

initialized at $\overrightarrow{\mathbf{U}}_0^{\mathrm{Bri}} \sim p_{\mathrm{fix}}$. The drift term $\mathbf{h}(t, \cdot)$ is typically affine: $\mathbf{h}(x, t) = -\lambda(t)x$, then the transition kernel $p_t(\overrightarrow{\mathbf{U}}_t^{\mathrm{Bri}}.|\overrightarrow{\mathbf{U}}_0^{\mathrm{Bri}})$ is Gaussian $\mathcal{N}(l_t, \Sigma_t)$ (Sarkka & Solin, 2019) [4], where the mean l_t and variance Σ_t evolve according to:

$$\frac{\mathrm{d}l_t}{\mathrm{d}t} = -\lambda(t)l_t, \quad l_0 = 0 \tag{12}$$

$$\frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} = -2\lambda(t)\Sigma_t + g(t)^2 I, \quad \Sigma_0 = \sigma^2 I. \tag{13}$$

Distribution of Bridge Process:

$$p_t^{\mathrm{Bri}}(\overrightarrow{\mathbf{U}}_t^{\mathrm{Bri}}) = \mathcal{N}(\overrightarrow{\mathbf{U}}_t^{\mathrm{Bri}}|0, \kappa_t I) \tag{14}$$

where $\kappa_t \triangleq \left(\int_0^t g(r)^2 e^{\int_0^r \lambda(s)\mathrm{d}s}\mathrm{d}r + \sigma^2\right) e^{-\int_0^t \lambda(s)\mathrm{d}s}$.

Reverse Process and SDE:

$$\begin{aligned}\mathrm{d}\overleftarrow{\mathbf{U}}_t^{\mathrm{Bri}} =& g(T-t)^2 \nabla \ln\left(p_{T-t}^{\mathrm{Bri}}\left(\overleftarrow{\mathbf{U}}_t^{\mathrm{Bri}}\right)\right)\mathrm{d}t - \mathbf{h}(T-t, \overleftarrow{\mathbf{U}}_t^{\mathrm{Bri}})\mathrm{d}t \\ &+ g(T-t)\mathrm{d}W_t,\end{aligned} \tag{15}$$

initialized at $\overleftarrow{\mathbf{U}}_0^{\mathrm{Bri}} \sim p_T^{\mathrm{Bri}}$.

Gradient of Log-Likelihood:

$$\nabla \ln\left(p_{T-t}^{\mathrm{Bri}}\left(\overleftarrow{\mathbf{U}}_t^{\mathrm{Bri}}\right)\right) = -\frac{\overleftarrow{\mathbf{U}}_t^{\mathrm{Bri}}}{\kappa_{T-t}}. \tag{16}$$

For the first term in Eq. 10, according to Girsanov Theorem (Oksendal, 2013) [2], we have

$$\mathbb{E}_{\mathcal{P}^\phi} \log \frac{\mathrm{d}\mathcal{P}^\phi}{\mathrm{d}\mathcal{P}^{\mathrm{Bri}}} = \mathrm{KL}\left(p_{\mathrm{fix}} \| p_T^{\mathrm{Bri}}\right) + \mathrm{KL}(\mathcal{Q}^\phi(\cdot|\overleftarrow{\mathbf{U}}_0^\phi) || \mathcal{Q}(\cdot|\overleftarrow{\mathbf{U}}_0^{\mathrm{Bri}})) \quad (17)$$

where

$$\begin{aligned} &\mathrm{KL}(\mathcal{Q}^\phi(\cdot|\overleftarrow{\mathbf{U}}_0^\phi) || \mathcal{Q}(\cdot|\overleftarrow{\mathbf{U}}_0^{\mathrm{Bri}})) \\ =&\frac{1}{2}\int_0^T \mathbb{E}_{\mathcal{Q}^\phi}\, g(T-t)^2 \left\| \frac{\overleftarrow{\mathbf{U}}_t^\phi}{\kappa_{T-t}} + \boldsymbol{s}_{\boldsymbol{\phi}}(T-t, \overleftarrow{\mathbf{U}}_t^\phi) \right\|_2^2 \mathrm{d}t \end{aligned} \quad (18)$$

For the second term, since $\mathcal{P}$ and $\mathcal{P}^{\mathrm{Bri}}$ have the same dynamic system τ except for different initial values, we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{P}^{\phi}} \log \frac{\mathrm{d}\mathcal{P}^{\mathrm{Bri}}}{\mathrm{d}\mathcal{P}} &= \mathbb{E}_{\mathcal{P}^{\phi}} \log \frac{\mathcal{P}^{\mathrm{Bri}}\left(\tau|\cdot\right) p_0^{\mathrm{Bri}}\left(\cdot\right)}{\mathcal{P}\left(\tau|\cdot\right) p_0\left(\cdot\right)} \\
&= \mathbb{E}_{\mathcal{Q}_T^{\phi}} \log \frac{p_0^{\mathrm{Bri}}\left(\cdot\right)}{p_0\left(\cdot\right)} \qquad (19) \\
&= \mathbb{E}_{\mathcal{Q}_T^{\phi}} \log \frac{p_{\mathrm{fix}}}{q} \\
&= \mathbb{E}_{\mathcal{Q}_T^{\phi}} \log \frac{p_{\mathrm{fix}}}{p(\mathbf{y}|\cdot)p(\cdot)} + \log p(\mathbf{y})
\end{aligned}$$

A New Evidence Lower Bound

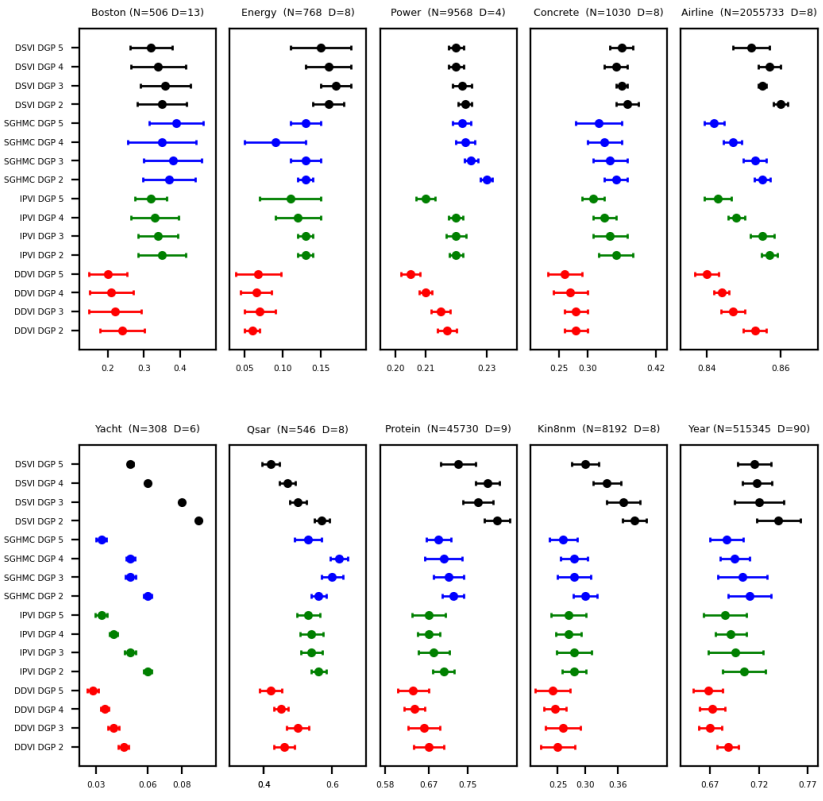
We define $l_1(\phi) = \mathbb{E}_{\mathcal{P}^\phi} \log \frac{\mathrm{d}\mathcal{P}^\phi}{\mathrm{d}\mathcal{P}^{\mathrm{Bri}}}$. Then, we obtain the following variational lower bound $l(\phi)$:

$$
\begin{aligned}
\log p(\mathbf{y}) &= \mathrm{KL}(\mathcal{P}^\phi \| \mathcal{P}) - l_1(\phi) - \mathbb{E}_{\mathcal{Q}_T^\phi} \log \frac{p_{\mathrm{fix}}}{p(\mathbf{y}|\cdot)p(\cdot)} \\
&= \mathrm{KL}(\mathcal{P}^\phi \| \mathcal{P}) - l_1(\phi) - \mathbb{E}_{\mathcal{Q}_T^\phi} \log p_{\mathrm{fix}} \\
&\quad + \mathbb{E}_{\mathcal{Q}_T^\phi} \log p(\cdot) + \mathbb{E}_{\mathcal{Q}_T^\phi, \mathbf{F}_1, \ldots, \mathbf{F}_L} \log p(\mathbf{y}|\mathbf{F}_L) \\
&\geqslant \mathbb{E}_{\mathcal{Q}_T^\phi} \log p(\cdot) + \mathbb{E}_{\mathcal{Q}_T^\phi, \mathbf{F}_1, \ldots, \mathbf{F}_L} \log p(\mathbf{y}|\mathbf{F}_L) - l_1(\phi) \\
&\quad - \mathbb{E}_{\mathcal{Q}_T^\phi} \log p_{\mathrm{fix}} \\
&= l(\phi).
\end{aligned}
\tag{20}
$$

Input: training data $\mathbf{X}, \mathbf{y}$ mini-batch size B
Initialize diffusion coefficient h, g, all DGP hyperparameters γ
repeat
  for $t_s = 0$ **to** $T-1$ **do**
    Draw $\boldsymbol{\epsilon}_{t_s} \sim \mathcal{N}(0, I)$ and set $\mathbf{U}_{t_s+1} =$
    $\mathbf{U}_{t_s} - \mathbf{h}(\mathbf{U}_{t_s}, T-t_s) + g(T-t_s)^2 s_\phi(T-t_s, \mathbf{U}_{t_s}) + g(T-t)\boldsymbol{\epsilon}_{t_s}$
    Compute $\kappa_{T-(t_s+1)}$ by Eq. (14) and set
    $l_{t_s+1} = l_{t_s} + g(T-(t_s+1))^2 \| \frac{\mathbf{U}_{t_s+1}}{\kappa_{T-(t_s+1)}} + \boldsymbol{s}_{\boldsymbol{\phi}}(T-(t_s+1), \mathbf{U}_{t_s+1}) \|_2^2$
  end for
  Sample mini-batch indices $I \subset \{1, \ldots, N\}$ with $|I| = B$ and set
  $\{\mathbf{U}_{\ell,1}, ..., \mathbf{U}_{\ell,D}\}_{\ell=1}^{L} = \mathbf{U}_T$
  for $\ell = 1$ **to** L **do**
    Draw $\epsilon_{\ell,d} \sim \mathcal{N}(0, I)$ and calculate $\mathbf{F}_{\ell,d} =$
    $\mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{Z}_\ell} \mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1} \mathbf{U}_{\ell,d} + \sqrt{\mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{F}_{\ell-1}} - \mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{z}_\ell} \mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1} \mathbf{K}_{\mathbf{Z}_\ell\mathbf{F}_{\ell-1}}} \boldsymbol{\epsilon}_{\ell,d}$
  end for
  Set $l(\phi, \gamma) = -\log p_{\text{fix}}(\mathbf{U}_T) + \log p(\mathbf{U}_T) + \frac{N}{B} \log p(\mathbf{y}_I \mid \mathbf{F}_L) -$
  $\mathrm{KL}(p_{\text{fix}} \| \mathcal{N}(0, \kappa_T)) - \frac{1}{2} l_T$
  Make a gradient descent update of $l(\phi, \gamma)$
until ϕ, γ converge

Experiments

Regression test RMSE.



Regression test mean NLL.

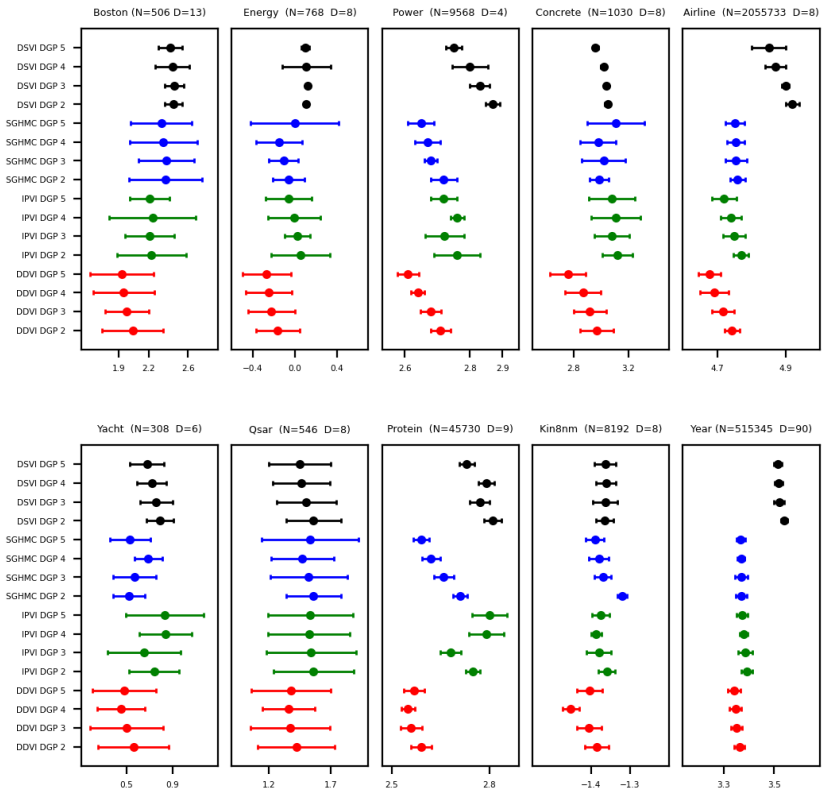


Image Dataset Classification

Table 1. Mean test accuracy (%) and training details achieved by DSVI, SGHMC, IPVI and DDVI (ours) DGP model for three image classification datasets. Results are shown for 3 and 4 layers as indicated, and runtime is given per iteration.

Data Set	Model	Time3	Iter3	Acc3	Time4	Iter4	Acc4
MNIST	DSVI	0.34s	20K	97.17	0.54s	20K	97.41
	IPVI	0.49s	20K	97.58	0.62s	20K	97.80
	SGHMC	1.14s	20K	97.25	1.22s	20K	97.55
	DDVI	0.38s	20K	**98.84**	0.50s	20K	**99.01**
Fashion	DSVI	0.34s	20K	87.45	0.50s	20K	87.99
	IPVI	0.48s	20K	88.23	0.61s	20K	88.90
	SGHMC	1.21s	20K	86.88	1.25s	20K	87.08
	DDVI	0.40s	20K	**90.36**	0.55s	20K	**90.85**
CIFAR-10	DSVI	0.43s	20K	91.47	0.66s	20K	91.79
	IPVI	0.62s	20K	92.79	0.78s	20K	93.52
	SGHMC	8.04s	20K	92.62	8.61s	20K	92.94
	DDVI	0.45s	20K	**95.23**	0.69s	20K	**95.56**

Large-Scale Dataset Classification.

Table 2. Test AUC values for large-scale classification datasets. Uses random 90% / 10% training and test splits.

		SUSY	HIGGS
	N	5,500,000	11,000,000
	D	18	28
DSVI $M = 128$	$L = 2$	0.876	0.830
	$L = 3$	0.877	0.837
	$L = 4$	0.878	0.841
	$L = 5$	0.878	0.846
IPVI $M = 128$	$L = 2$	0.879	0.843
	$L = 3$	0.882	0.847
	$L = 4$	0.883	0.850
	$L = 5$	0.883	0.852
SGHMC $M = 128$	$L = 2$	0.879	0.842
	$L = 3$	0.881	0.846
	$L = 4$	0.883	0.850
	$L = 5$	0.884	0.853
DDVI $M = 128$	$L = 2$	**0.883**	**0.849**
	$L = 3$	**0.885**	**0.852**
	$L = 4$	**0.887**	**0.856**
	$L = 5$	**0.886**	**0.857**

Unsupervised Learning for Data Recovery Task

Table 3. Mean RMSE and NLL achieved by DSVI, SGHMC, IPVI and DDVI (ours) GPLVM model for data recovery task. Standard deviation is shown in parentheses. Runtime is given per iteration.

Data Set	Model	Time	Iter	RMSE	NLL
Frey Faces	DSVI	0.32s	20K	8.32 (0.2)	1.49 (0.02)
	IPVI	0.42s	20K	7.91 (0.4)	1.33 (0.02)
	SGHMC	1.13s	20K	7.95 (0.3)	1.36 (0.03)
	DDVI	0.36s	20K	**7.64** (0.2)	**1.17** (0.01)

References I


Andreas Damianou and Neil Lawrence.
Deep Gaussian processes.
In *Conference on Artificial Intelligence and Statistics*, pages 207–215, 2013.


Bernt Oksendal.
Stochastic differential equations: an introduction with applications.
Springer Science & Business Media, 2013.


Hugh Salimbeni and Marc Deisenroth.
Doubly stochastic variational inference for deep Gaussian processes.
In *Conference on Neural Information Processing Systems*, pages 4588–4599, 2017.

References II


Simo Särkkä and Arno Solin.
Applied stochastic differential equations, volume 10.
Cambridge University Press, 2019.


M. K. Titsias.
Variational model selection for sparse Gaussian process regression.
Technical report, School of Computer Science, University of Manchester, 2009.


Haibin Yu, Yizhou Chen, Bryan Kian Hsiang Low, Patrick Jaillet, and Zhongxiang Dai.
Implicit posterior variational inference for deep gaussian processes.
Advances in neural information processing systems, 32, 2019.

Questions

Thank you!
Questions?