UQSay #54

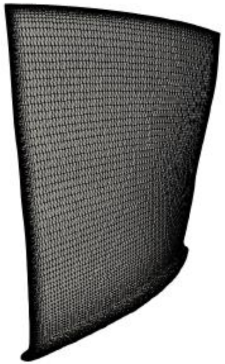
Quantitative performance evaluation of Bayesian neural networks (benchmark)

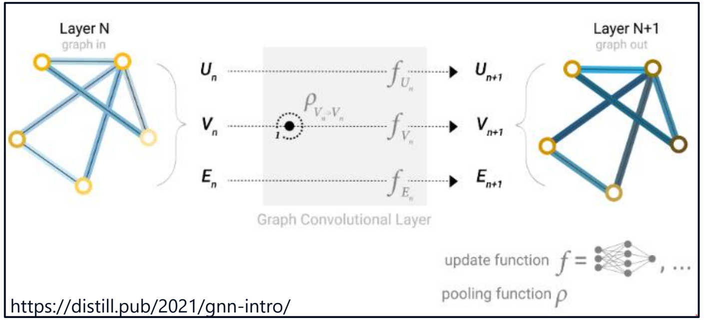
Brian Staber (Safran)
Sébastien Da Veiga (ENSAI)

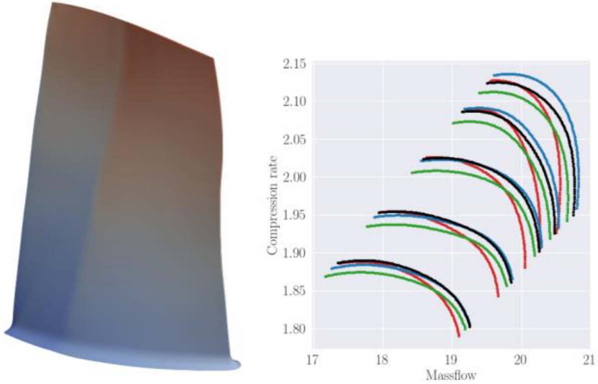
SAFRAN

Deep learning for computational engineering

Example in computational fluid dynamics







Input mesh | **Graph neural network** | **Predict fields and scalars**

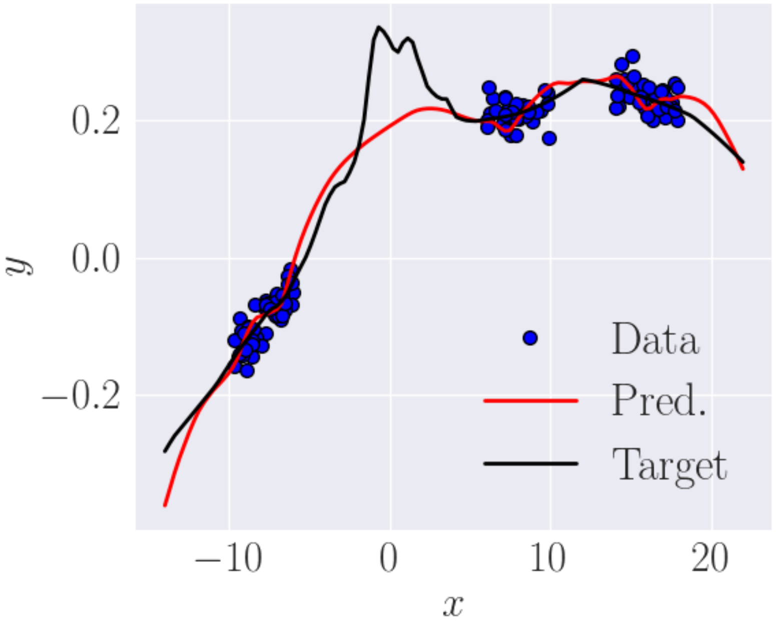
Deep learning

Supervised training

- Training dataset : $\mathcal{D} = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^{N}$
- Neural network $\hat{f}(\cdot; \mathbf{w})$ with parameters $\mathbf{w} \in \mathbb{R}^d$
- Train the network by minimizing a loss function

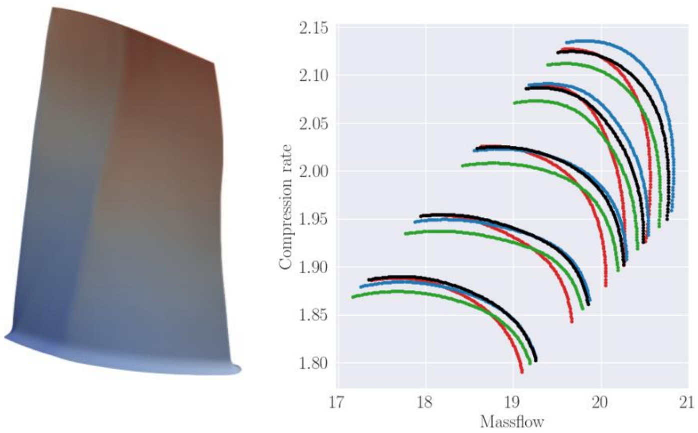
$$L(\mathbf{w}) = -\sum_{i=1}^{N} \log p(\mathbf{Y}_i | \mathbf{X}_i, \mathbf{w})$$

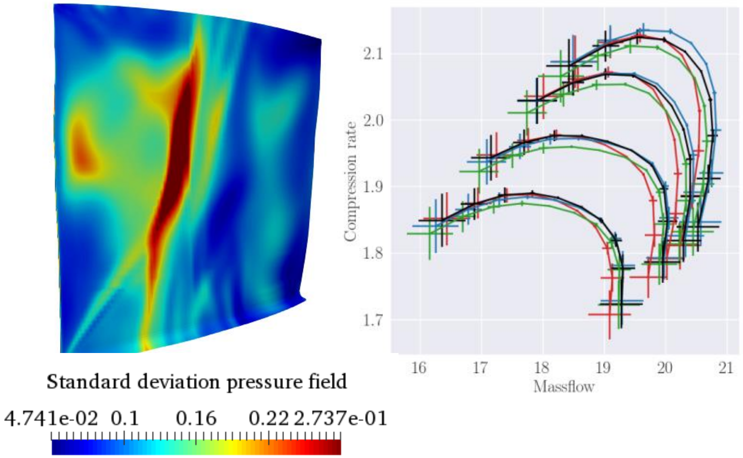
- Point estimate $\mathbf{w}_{\mathrm{MLE}}$, no predictive uncertainties



Predictive uncertainties

Example in computational fluid dynamics





Neural network : predictions

What we need : predictive uncertainties

Bayesian neural networks

Bayesian inference

- The observations take the form

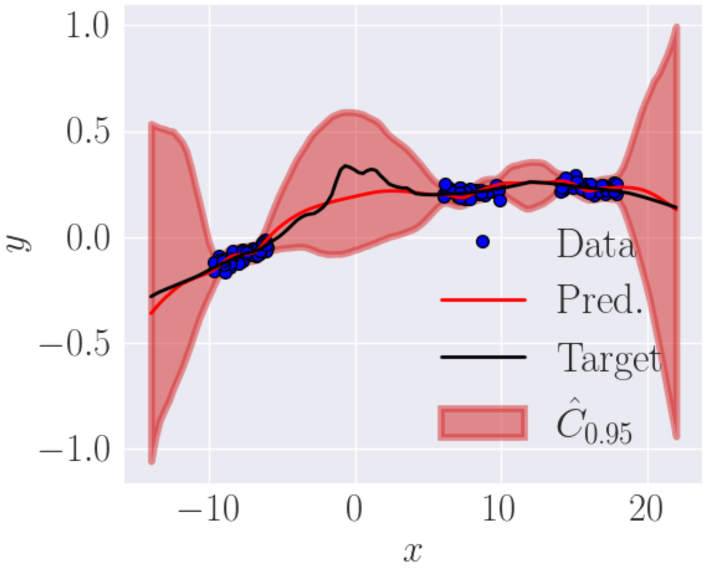
$$\mathbf{Y}_i = f(\mathbf{X}_i) + \boldsymbol{\varepsilon}_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2(\mathbf{X}_i)\mathbf{I}_M)$$

- Pick a prior distribution $p(\mathbf{w})$ over the parameters

- Deduce the posterior distribution using Bayes formula

$$p(\mathbf{w}|\mathbf{X},\mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})$$

$$p(\mathbf{Y}_i|\mathbf{X}_i,\mathbf{w}) = \mathcal{N}(\mathbf{Y}_i; \hat{f}(\mathbf{X}_i;\mathbf{w}), \sigma^2\mathbf{I}_M)$$



SAFRAN

Bayesian neural networks

Deep bayesian neural networks are challenging

- Prior distribution over the parameters difficult to choose
- Large datasets and possibly high-dimensional inputs/outputs
- High dimensional intractable posterior, possibly multimodal

Exact inference:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{Y}) = \int_{\mathbb{R}^d} p(\mathbf{y}|\mathbf{x}, \mathbf{w}) p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) d\mathbf{w}$$

Approximate inference:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{Y}) \approx \frac{1}{n} \sum_{i=1}^{n} p(\mathbf{y}|\mathbf{x}, \mathbf{w}_i) \,, \quad \mathbf{w}_i \sim q(\mathbf{w}) \approx p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$$

SAFRAN

Approximate inference

Sampling methods

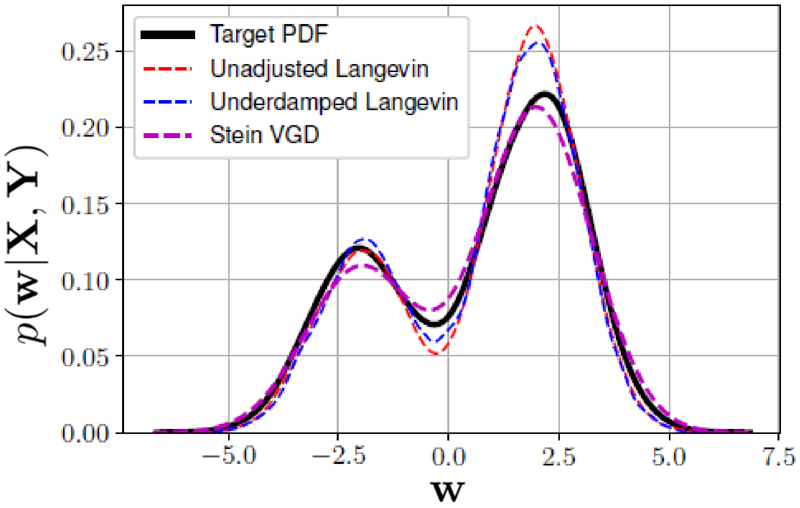
- Classical MCMC (HMC, NUTS, MALA, ...)
- Stochastic gradient MCMC (SGLD, SGHMC, ...)

Variational methods

- Classical VI (MFV, BBB, ...)
- Stein variational gradient descent
- Monte Carlo dropout

Gaussian approximations

- Laplace (diagonal or kronecker factorization matrix)
- Stochastic weight averaging Gaussian (SWAG)



Approximate inference

- Which methods generate valid confidence intervals ?
- Which methods provide the best approximations to the target posterior ?
- Do we really need a good approximation in the high-dimensional weight space ?
- Sensitivity with respect to hyperparameters ?
- Are there any similarities between some algorithms ?

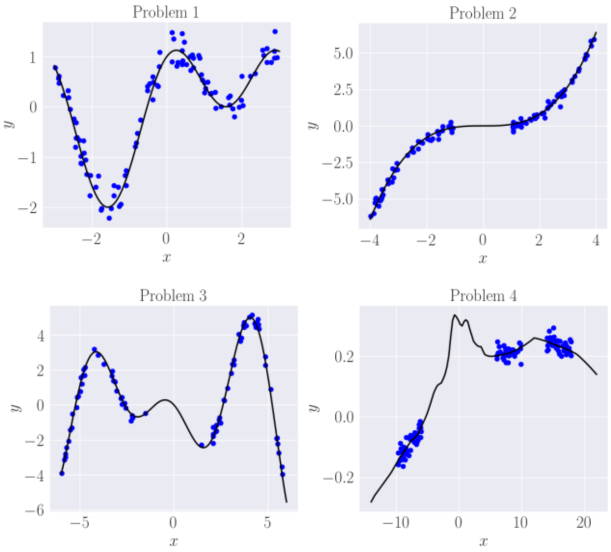
Benchmark setup

Approximation methods

- Hamiltonian Monte Carlo
- Stochastic gradient MCMC
- MC dropout
- Deep ensembles
- Laplace approximation, SWAG

Experiments

- 4 synthetic regression problems
- MLP networks



Evaluation metrics

- Coverage probabilities
- Prediction accuracy
- Distances between probability distributions

Approximation methods

Markov Chain Monte Carlo methods

Hamiltonian Monte Carlo

- Used as a reference
- 3 chains, 200 iterations and 10,000 leapfrog steps
- Step size chosen to get appropriate accept rates
- Requires the full gradient of the log-posterior
- HMC for BNNs studied by Izmaloiv et al., *What are Bayesian neural network posteriors really like?*

Stochastic gradient MCMC

- Metropolis-Hastings correction step omitted
- Stochastic gradient (mini batched)
- Computationally more efficient but introduces asymptotic bias
- 9 variants are considered

SAFRAN

Stochastic gradient MCMC

Stochastic differential equation [Ma et al. 2015, Nemeth et al. 2020]

$$d\mathbf{Z} = \frac{1}{2}\mathbf{b}(\mathbf{Z})\,dt + \sqrt{\mathbf{D}(\mathbf{Z})}\,d\mathbf{W}(t)\,, \qquad \mathbf{Z}_t = (\mathbf{w}_t, \mathbf{r}_t)$$

$$\mathbf{b}(\mathbf{Z}) = -(\mathbf{D}(\mathbf{Z}) + \mathbf{Q}(\mathbf{Z}))\nabla H(\mathbf{Z}) + \Gamma(\mathbf{Z})\,, \quad \Gamma_i(\mathbf{Z}) = \sum_j \frac{\partial}{\partial \mathbf{Z}_j}(\mathbf{D}_{ij}(\mathbf{Z}) + \mathbf{Q}_{ij}(\mathbf{Z}))$$

Energy function

$$H(\mathbf{Z}) = U(\mathbf{w}) + K(\mathbf{r})\,,$$

$$U(\mathbf{w}) = -\log(p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})) = -\sum_{i=1}^{N} \log(p(\mathbf{Y}_i|\mathbf{X}_i,\mathbf{w})) - \log(p(\mathbf{w}))$$

Stochastic gradient MCMC

Euler explicit scheme and mini-batched gradients

$$\mathbf{Z}^{k+1} = \mathbf{Z}^k - \epsilon\left((\mathbf{D}(\mathbf{Z}^k) + \mathbf{Q}(\mathbf{Z}^k))\widehat{\nabla} H(\mathbf{Z}^k) + \Gamma(\mathbf{Z}^k)\right) + \sqrt{2\epsilon \mathbf{D}(\mathbf{Z}^k)}\,\Delta \mathbf{W}^{k+1}$$

$$\widehat{\nabla} U(\mathbf{w}) = -\frac{N}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \nabla U_i(\mathbf{w}) - \nabla \log(p(\mathbf{w}))$$

Table 1. A list of popular SGMCMC algorithms highlighting how they fit within the general stochastic differential equation framework (9) and (10).

Algorithm	ζ	$H(\zeta)$	$\mathbf{D}(\zeta)$	$\mathbf{Q}(\zeta)$
SGLD	$\boldsymbol{\theta}$	$U(\boldsymbol{\theta})$	$\mathbf{I}$	$\mathbf{0}$
SGRLD	$\boldsymbol{\theta}$	$U(\boldsymbol{\theta})$	$G(\boldsymbol{\theta})^{-1}$	$\mathbf{0}$
SGHMC	$(\boldsymbol{\theta}, \boldsymbol{\rho})$	$U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\rho}^\top\boldsymbol{\rho}$	$\begin{pmatrix} 0 & 0 \\ 0 & \mathbf{C} \end{pmatrix}$	$\begin{pmatrix} 0 & -\mathbf{I} \\ \mathbf{I} & 0 \end{pmatrix}$
SGRHMC	$(\boldsymbol{\theta}, \boldsymbol{\rho})$	$U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\rho}^\top\boldsymbol{\rho}$	$\begin{pmatrix} 0 & 0 \\ 0 & G(\boldsymbol{\theta})^{-1} \end{pmatrix}$	$\begin{pmatrix} 0 & -G(\boldsymbol{\theta})^{-1/2} \\ G(\boldsymbol{\theta})^{-1/2} & 0 \end{pmatrix}$
SGNHT	$(\boldsymbol{\theta}, \boldsymbol{\rho}, \eta)$	$U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\rho}^\top\boldsymbol{\rho} + \frac{1}{2d}(\eta - A)^2$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & A\cdot\mathbf{I} & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -\mathbf{I} & 0 \\ \mathbf{I} & 0 & \boldsymbol{\rho}^\top/d \\ 0 & -\boldsymbol{\rho}^\top/d & 0 \end{pmatrix}$

NOTE: Most of the terms are defined in the text, except: $\mathbf{C} \succeq h\mathbf{V}(\boldsymbol{\theta})$, which is a positive semidefinite matrix; $G(\boldsymbol{\theta})$ is the Fisher information metric; A is a tuning parameter for SGNHT.

Nemeth et al. 2020

Stochastic gradient MCMC

Vanilla SGMCMC

- SGLD [Welling & The, 2011]

$$k \geq 0\,, \quad \mathbf{w}^{k+1} = \mathbf{w}^k - \epsilon_k \widehat{\nabla} U(\mathbf{w}^k) + \sqrt{2\epsilon_k}\Delta\mathbf{W}^{k+1}$$

- SGHMC [Chen et al, 2014]

$$k \geq 0 : \begin{cases} \mathbf{w}^{k+1} = \mathbf{w}^k + \mathbf{v}^k\,, \\ \mathbf{v}^{k+1} = (1-\alpha)\mathbf{v}^k - \epsilon_k \widehat{\nabla} U(\mathbf{w}^k) + \sqrt{2\alpha\epsilon_k}\Delta\mathbf{W}^{k+1} \end{cases}$$

Stochastic gradient MCMC

Vanilla SGMCMC

- SGLD [Welling & The, 2011]
- SGHMC [Chen et al, 2014]

Variance reduction

- SGLD-CV [Baker et al, 2019]
- SGHMC-CV

Updated variance reduction

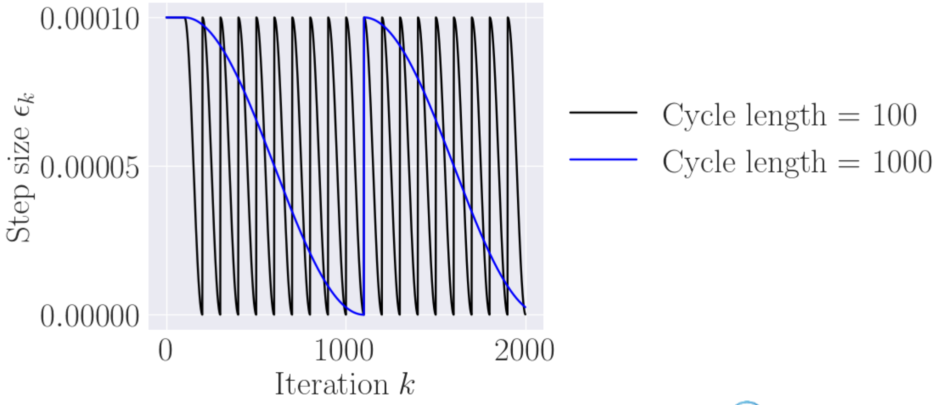
- SGLD-SVRG [Dubey et al, 2016]
- SGHMC-SVRG

With preconditioning

- pSGLD [Li et al, 2016]

Cyclical scheduler

- C-SGHMC [Zhang et al, 2019]
- C-SGLD

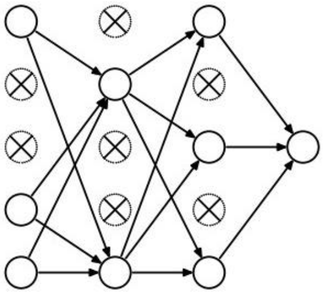


SAFRAN

MC Dropout and deep ensembles

Monte Carlo dropout [Gal et al., 2016]

- Train a neural network with dropout layers
- Output features of each layer are randomly dropped
- Keep dropout enabled during predictions



Deep ensembles [Lakshminarayanan et al., 2017]

- Train N networks independently
- Random initializations
- Aggregate the predictions

Gaussian approximations

Laplace approximation (LA-KFAC)

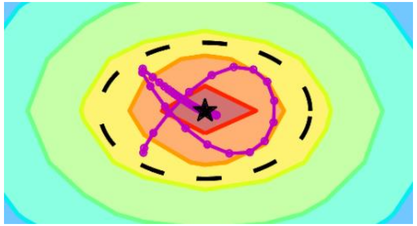
- Laplace approximation of the posterior
- Compute a MAP estimate by training the network
- Kronecker factored log likelihood Hessian approximation

$$p(\mathbf{w}|\mathbf{X},\mathbf{Y}) \approx \mathcal{N}(\mathbf{w}|\mathbf{w}_{\mathrm{MAP}}, \Sigma_{\mathrm{KFAC}})$$

Stochastic Weight Averaging Gaussian (SWAG)

- Compute a MAP estimate by training the network
- Run SGD with a high step size and collect values of the parameters
- Construct a Gaussian approximation to the posterior

$$p(\mathbf{w}|\mathbf{X},\mathbf{Y}) \approx \mathcal{N}(\mathbf{w}|\mathbf{w}_{\mathrm{SWAG}}, \Sigma_{\mathrm{SWAG}})$$



Maddox et al., 2019

Approximation methods : summary

Approximation methods (14)

- Hamiltonian Monte Carlo (used as a reference)
- Stochastic gradient MCMC (9 variants)
- MC dropout
- Deep ensembles
- Laplace approximation
- SWAG

Hyperparameters

- 10 step sizes (or learning rates)
- 5 dropout rates
- 3 cycle lengths in cyclical SGMCMC

Evaluation metrics

Considered metrics

Coverage probabilities

- Marginal coverage
- Conditional coverage

Distances between probability distributions

- Distance to the HMC reference (weight and function space)
- Distance to the target posterior distribution (weight space)

Similarities between the algorithms

- Pairwise distances (weight and function space)
- Multidimensional scaling

Coverage probabilities

Prediction interval coverage probability

- Yao et al. (2019) studies the prediction interval coverage probability (**PICP**)

$$\mathbb{P}\{\mathbf{Y}^\star \in \hat{C}_\alpha^{\mathcal{D}}(\mathbf{X}^\star)|\mathcal{D}\} \geq 1 - \alpha$$

where the probability is taken **only** over the test data $(\mathbf{X}^\star, \mathbf{Y}^\star)$.

- Variability related to the training dataset not taken into account
- Stronger coverage notions exist
- **Marginal coverage and conditional coverage**

Coverage probabilities

Marginal coverage probability (MCP)

$$\mathbb{P}\{\mathbf{Y}^\star \in \hat{C}_\alpha^{\mathcal{D}}(\mathbf{X}^\star)\} \geq 1-\alpha$$

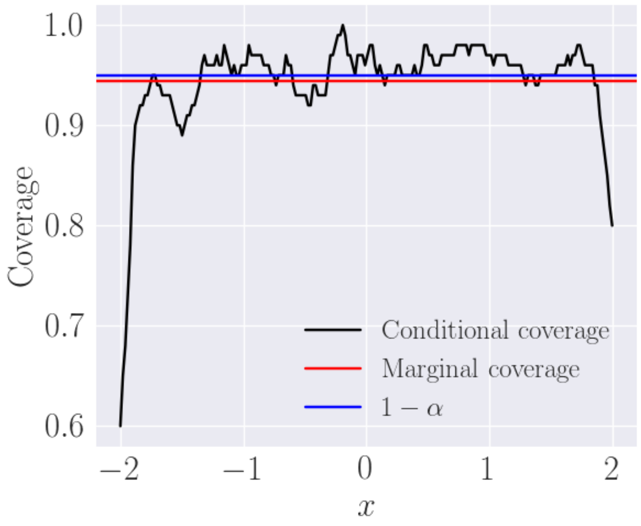
Probability taken over **both** the training and test data.

Conditional coverage probability (CCP)

$$\mathbb{P}\{\mathbf{Y}^\star \in \hat{C}_\alpha^{\mathcal{D}}(\mathbf{X}^\star)|\mathbf{X}^\star = \mathbf{x}\} \geq 1-\alpha$$

for almost all $\mathbf{x} \in \mathcal{X}$, probability taken over the training data.

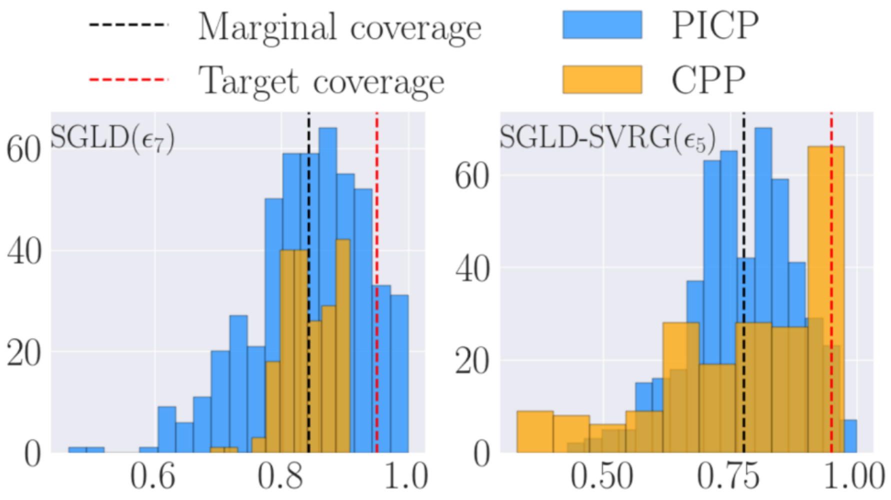
Target coverage: 0.95



Coverage probabilities

Example for two approximation methods

- Marginal coverage far from the target level
- Conditional coverage is pessimistic as well
- PICP exhibits large variability, may lead to wrong conclusions

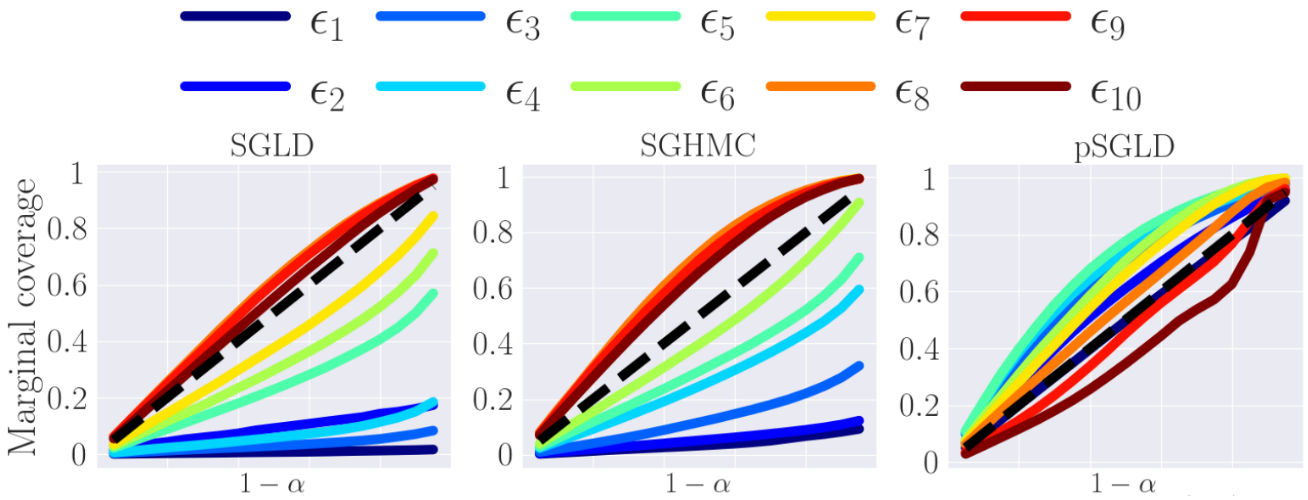


SAFRAN

Marginal and conditional coverage

For a given experiment:

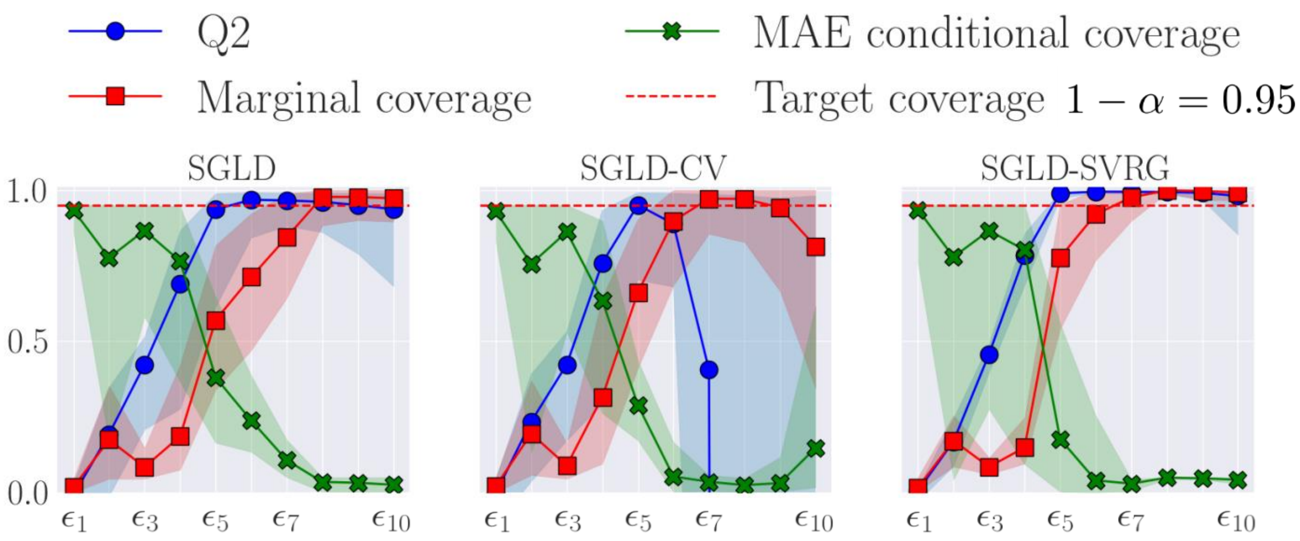
- Generate $N_{\mathcal{D}}$ independent training datasets $\mathcal{D}_1, \ldots, \mathcal{D}_{N_{\mathcal{D}}}$
- Generate a test dataset $\mathcal{D}^\star = \{(\mathbf{X}_i^\star, \mathbf{Y}_i^\star)\}_{i=1}^{N^\star}$



Marginal and conditional coverage

For a given experiment:

- Generate $N_{\mathcal{D}}$ independent training datasets $\mathcal{D}_1, \ldots, \mathcal{D}_{N_{\mathcal{D}}}$
- Generate a test dataset $\mathcal{D}^{\star} = \{(\mathbf{X}_i^{\star}, \mathbf{Y}_i^{\star})\}_{i=1}^{N^{\star}}$



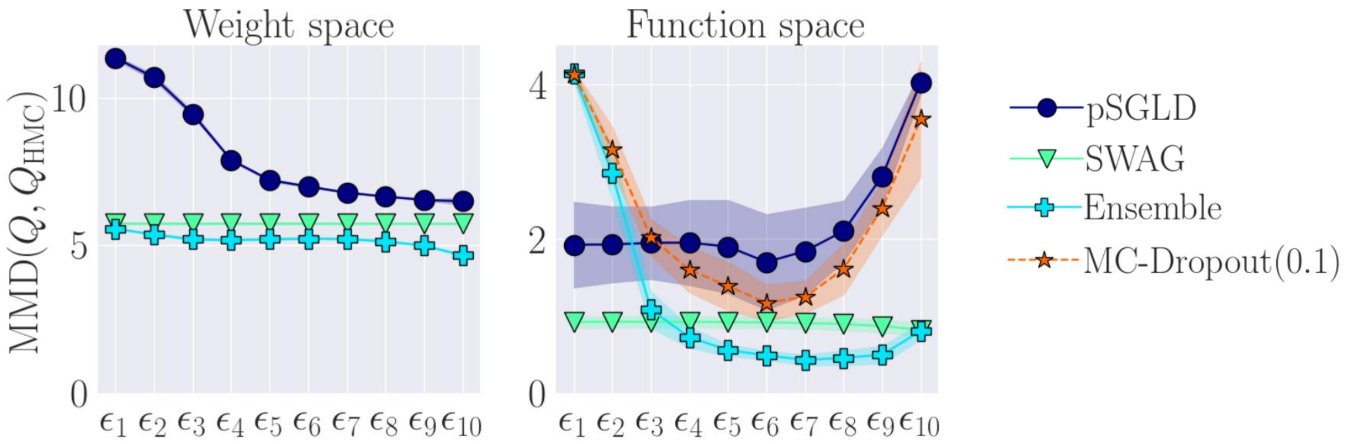
Distance to the HMC reference

Maximum mean discrepancy [Gretton et al., 2006]

- MMD distance between two probability measures

$$\mathrm{MMD}(\mathbb{Q}, \mathbb{Q}') = \|\mu_{\mathbb{Q}} - \mu_{\mathbb{Q}'}\|_{\mathcal{H}(k)}\,, \quad \mu_{\mathbb{Q}} = \int k(\cdot, \mathbf{w})\, d\mathbb{Q}$$

- Computed in weight and function spaces



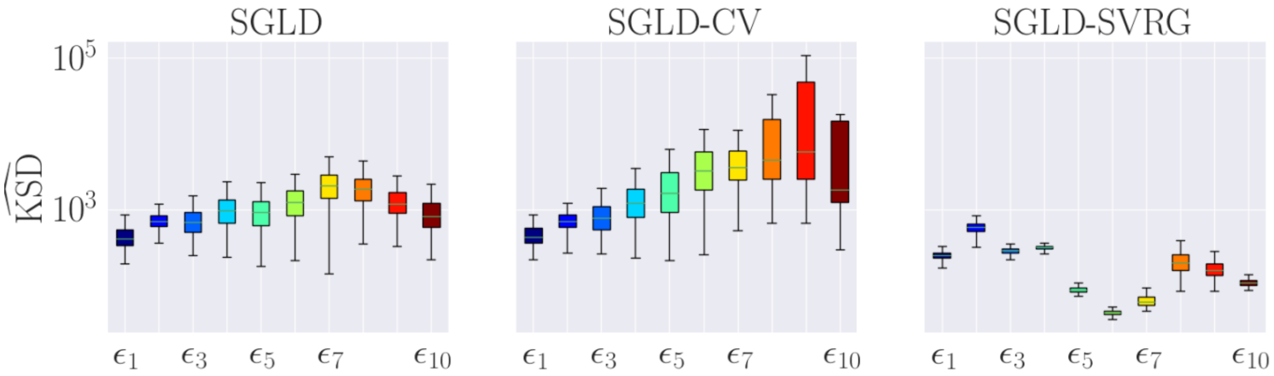
Distance to the target posterior

Kernelized Stein discrepancy [Gorham et al., 2015][Liu et al., 2016]

- Relies on Stein's discrepancy
- Only requires the knowledge of the score function $s_p(\mathbf{w}) = \nabla \log p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$

$$\mathrm{KSD}^2(\mathbb{P}, \mathbb{Q}) = \mathbb{E}[k_p(\mathbf{Z}, \mathbf{Z}')]\,, \quad \mathbf{Z} \sim \mathbb{Q}\,, \quad \mathbf{Z}' \sim \mathbb{Q}\,,$$

$$k_p(\mathbf{w}, \mathbf{w}') = \langle \nabla_{\mathbf{w}}, \nabla_{\mathbf{w}'} k(\mathbf{w}, \mathbf{w}') \rangle + \langle s_p(\mathbf{w}), \nabla_{\mathbf{w}'} k(\mathbf{w}, \mathbf{w}') \rangle + \langle s_p(\mathbf{w}'), \nabla_{\mathbf{w}} k(\mathbf{w}, \mathbf{w}') \rangle + \langle s_p(\mathbf{w}), s_p(\mathbf{w}') \rangle k(\mathbf{w}, \mathbf{w}')$$



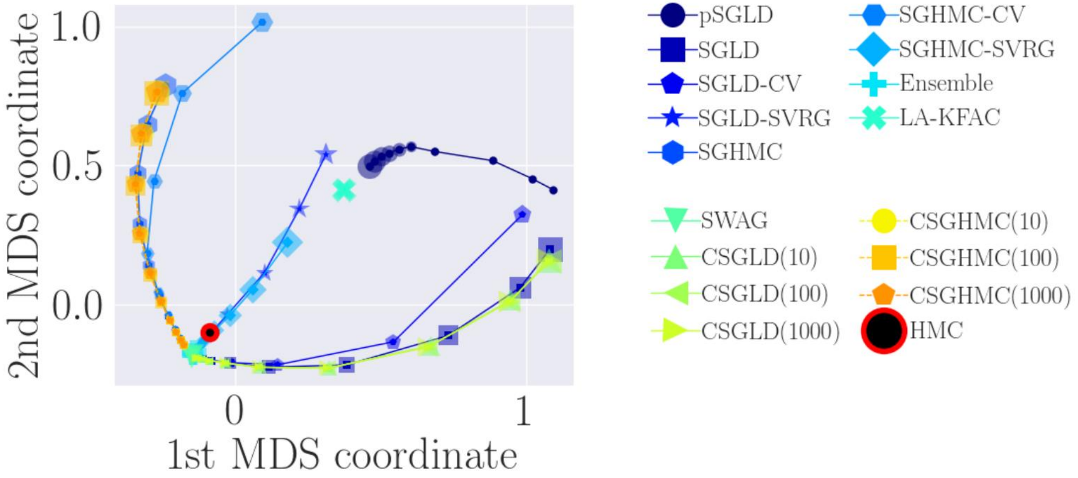
Similarities between the algorithms

Maximum mean discrepancy [Gretton et al., 2006]

- MMD distance between two probability measures

$$\mathrm{MMD}(\mathbb{Q}, \mathbb{Q}') = \|\mu_{\mathbb{Q}} - \mu_{\mathbb{Q}'}\|_{\mathcal{H}(k)}\ , \quad \mu_{\mathbb{Q}} = \int k(\cdot, \mathbf{w})\, d\mathbb{Q}$$

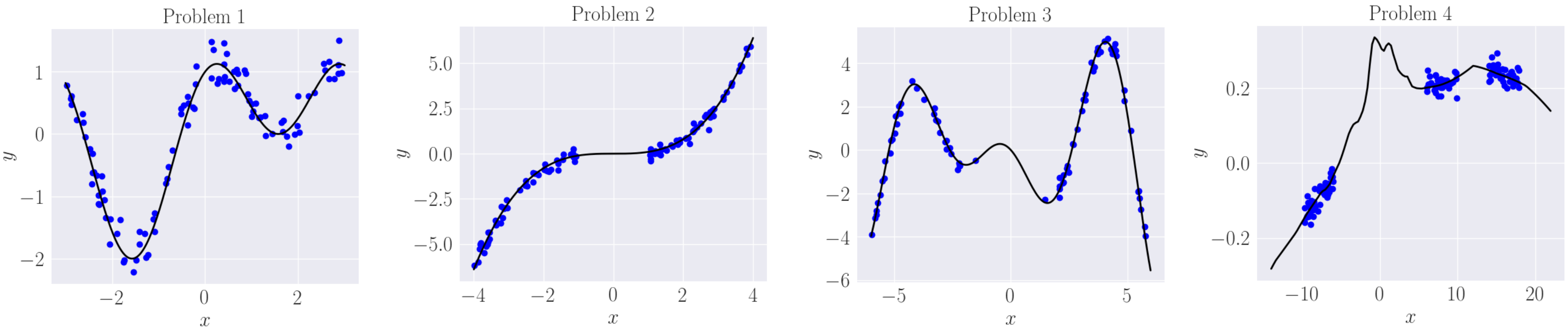
- Pairwise MMD distances and multidimensional scaling



Experiments

One-dimensional regression problems

Task	Latent function	σ	D	N	$N^\star$	OOD
AF#1	$\cos(2x) + \sin(x)$	0.2	1	100	200	✗
AF#2	$0.1x^3$	0.25	1	100	200	✓
AF#3	$-(1+x)\sin(1.2x)$	0.25	1	82	200	✓
AF#4	$\mathrm{MLP}(\cdot;\mathbf{w}),\ \mathbf{w} \sim \mathcal{N}(0, \mathbf{I})$	0.02	2	120	120	✓

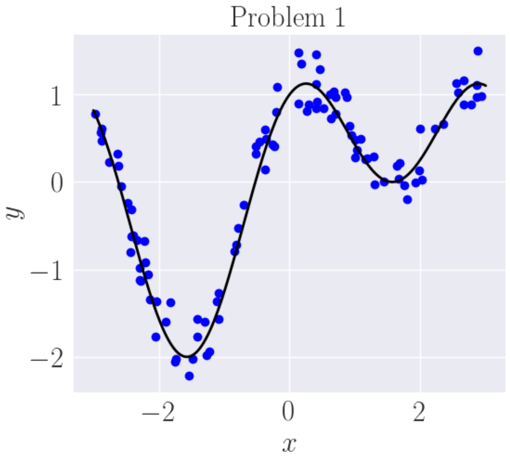


Regression problem #1

In the following slides

- Graphs of coverage probabilities
- Graphs of MMD distances
- Graphs of kernelized Stein discrepancies

- 14 approximations methods
- 10 step sizes $\epsilon_1, \ldots, \epsilon_{10}$
- 5 dropout rates for MC Dropout
- 3 cycle lengths for cyclical SGMCMC
- MLP network with 10k parameters



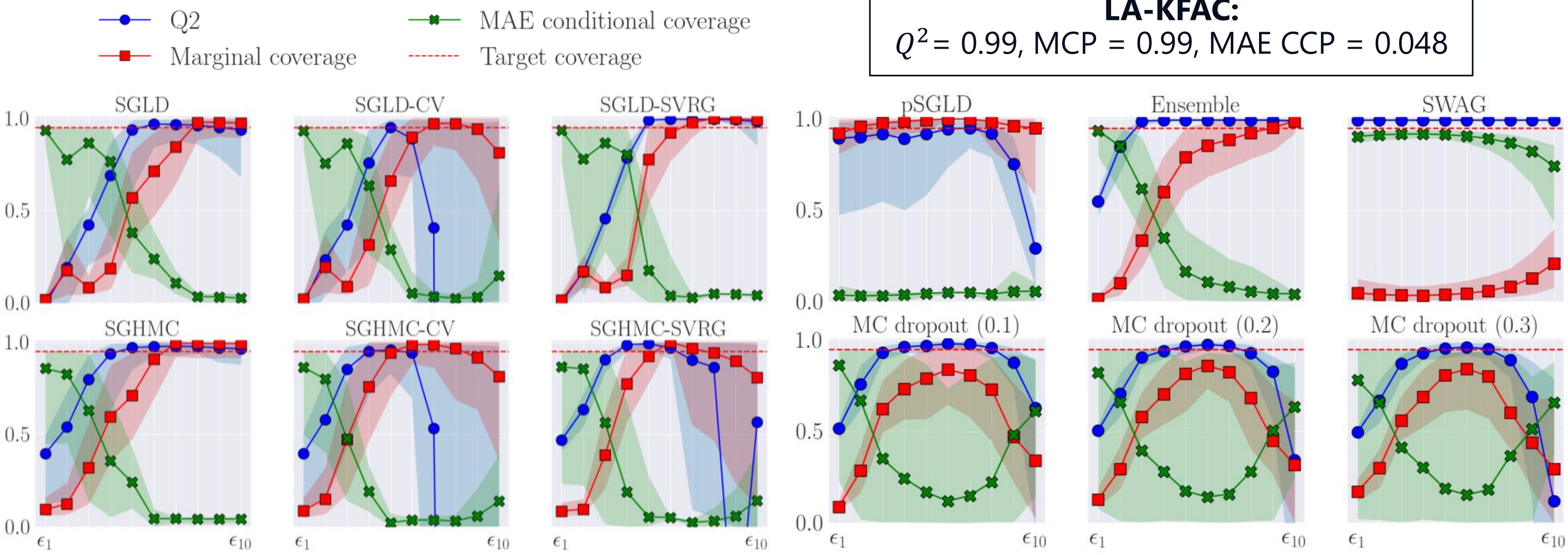
Coverage probabilities

- Estimated with 1000 datasets

Regression problem #1

Coverage probabilities for a 0.95 target and coefficients of determination Q^2

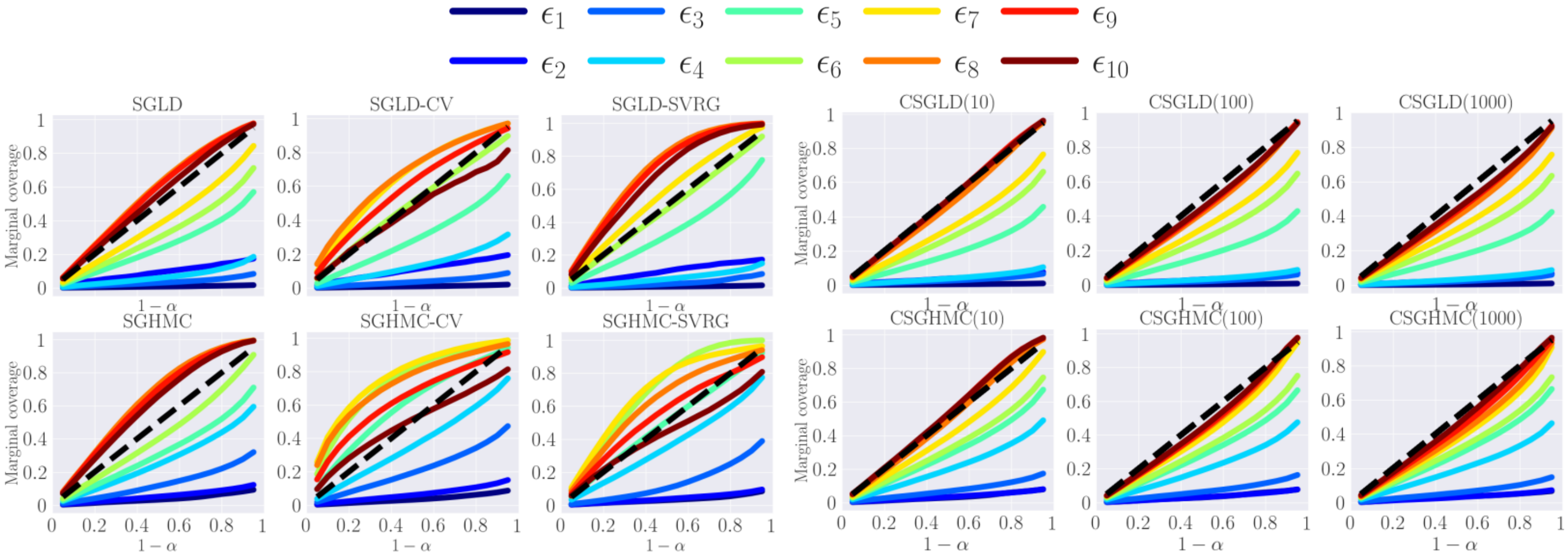
LA-KFAC:
$Q^2 = 0.99$, MCP = 0.99, MAE CCP = 0.048



Regression problem #1

- SGMCMC easily overshoots
- Highly sensitive w.r.t step size
- Cyclical step sizes reduce the coverages

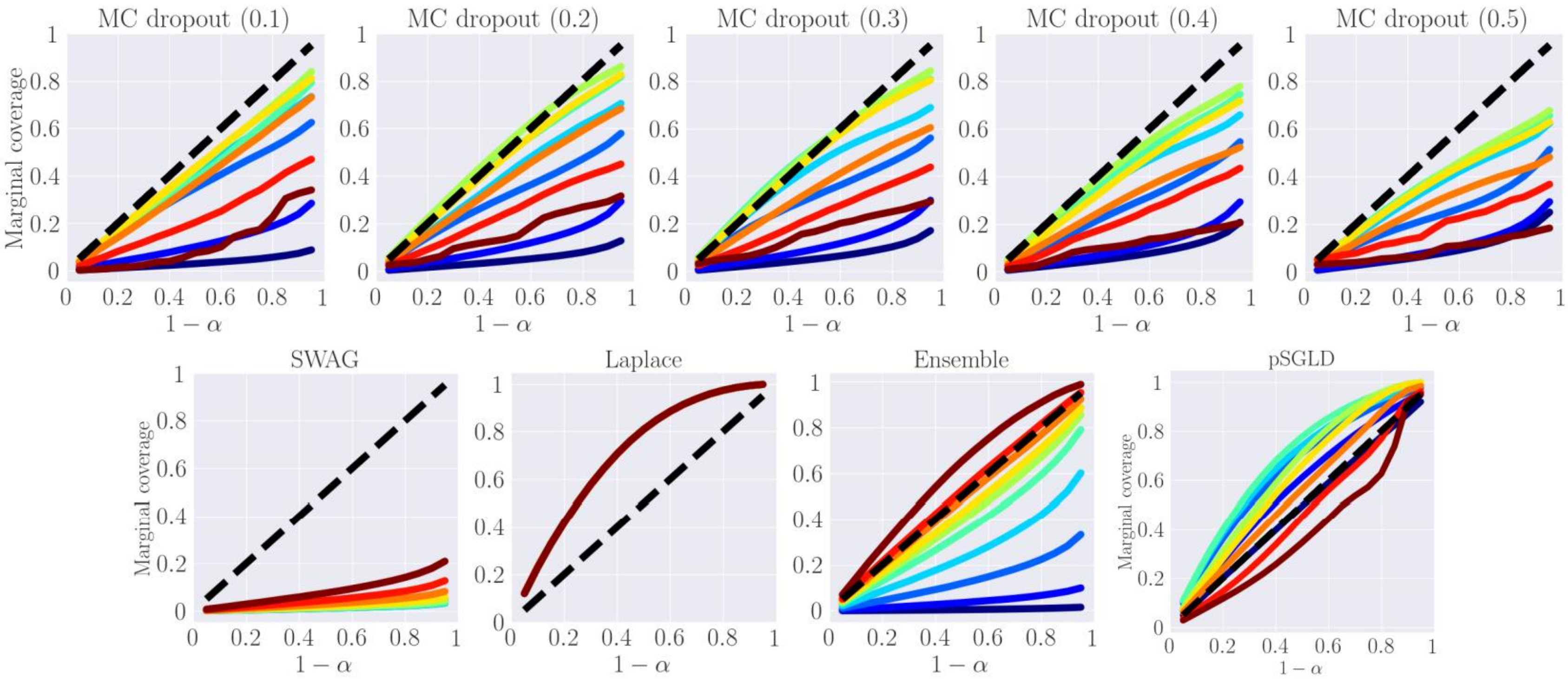
Graphs of the marginal coverage probabilities : SGMCMC and C-SGMCMC



Regression problem #1

- MC dropout and SWAG struggle
- LA-KFAC and pSGLD easily overestimate
- Deep ensembles seems promising

Graphs of the marginal coverage probabilities



SAFRAN

Regression problem #1

Best marginal coverage versus best prediction accuracy

Table 2. Best marginal coverage probabilities (MCP) and best Q^2 coefficients obtained with each algorithm.

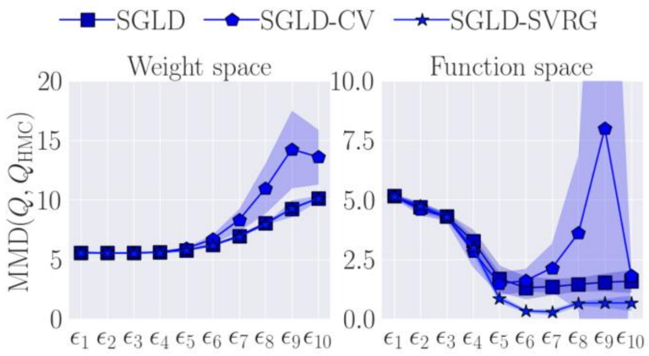
Method	Best MCP	Q^2	Best Q^2	MCP	Method	Best MCP	Q^2	Best Q^2	MCP
SGLD	0.97	0.93	0.96	0.71	CSGHMC(10)	0.97	0.97	0.97	0.89
SGLD-CV	**0.94**	-10^3	0.95	0.66	CSGHMC(100)	**0.94**	0.97	0.97	0.94
SGLD-SVRG	0.97	0.99	**0.99**	0.91	CSGHMC(1000)	**0.94**	0.96	0.97	0.93
SGHMC	0.99	0.96	0.97	0.99	pSGLD	**0.94**	0.29	0.95	0.99
SGHMC-CV	**0.94**	0.95	0.95	0.94	Deep ensemble	**0.95**	0.99	**0.99**	0.88
SGHMC-SVRG	**0.94**	0.86	**0.99**	0.92	SWAG	0.20	0.99	**0.99**	0.03
CSGLD(10)	**0.95**	0.96	0.96	0.66	MC Drop.(0.1)	0.83	0.98	**0.98**	0.83
CSGLD(100)	**0.94**	0.96	0.96	0.76	MC Drop.(0.2)	0.85	0.97	0.97	0.85
CSGLD(1000)	0.92	0.95	0.96	0.75	MC Drop.(0.3)	0.84	0.96	0.96	0.84

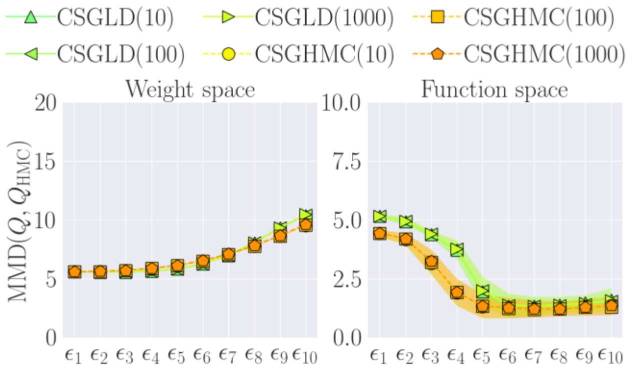
LA-KFAC:
Q^2 = 0.99, MCP = 0.99

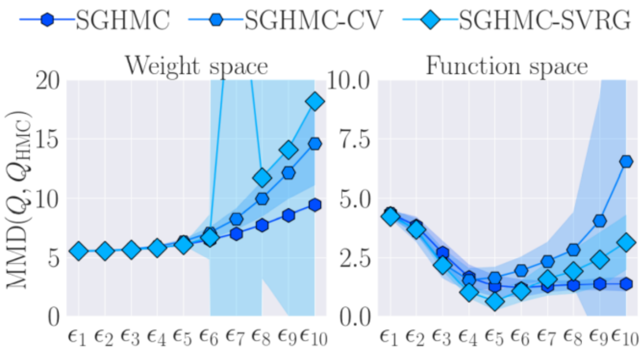
Regression problem #1

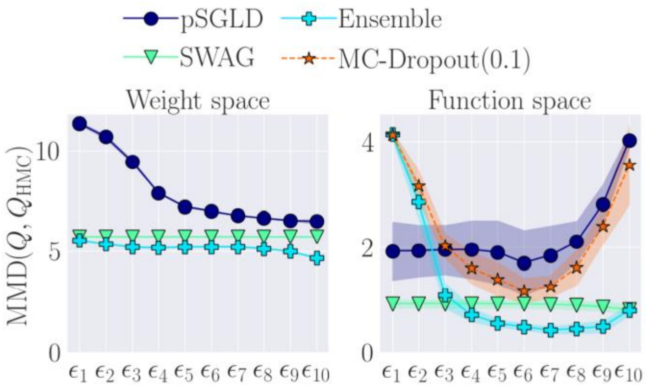
MMD distance to the HMC reference

- Different behavior weight / function space
- Lowest MMD in function space : SGMCMC-SVRG & Ensemble
- Deep ensembles seems promising
- CV methods are unstable









Regression problem #1

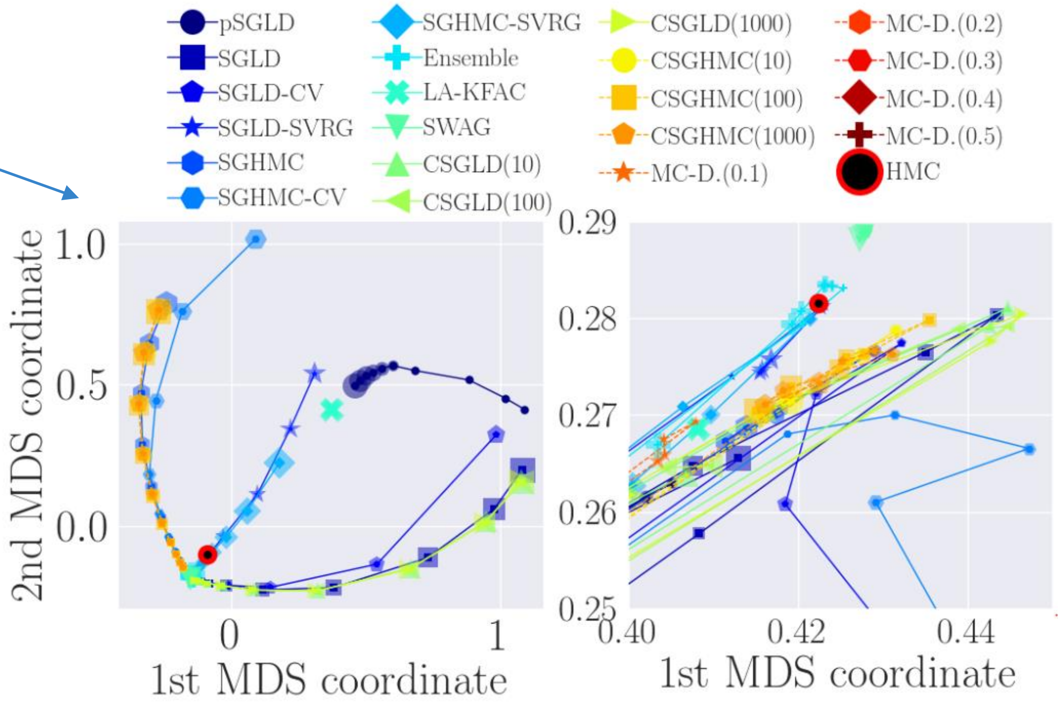
Similarities between the algorithms

- Structured similarities in weight space
- Messy similarities in function space
- SVRG & Ensemble closest to HMC

Weight space

Function space

Marker size proportional to the step size



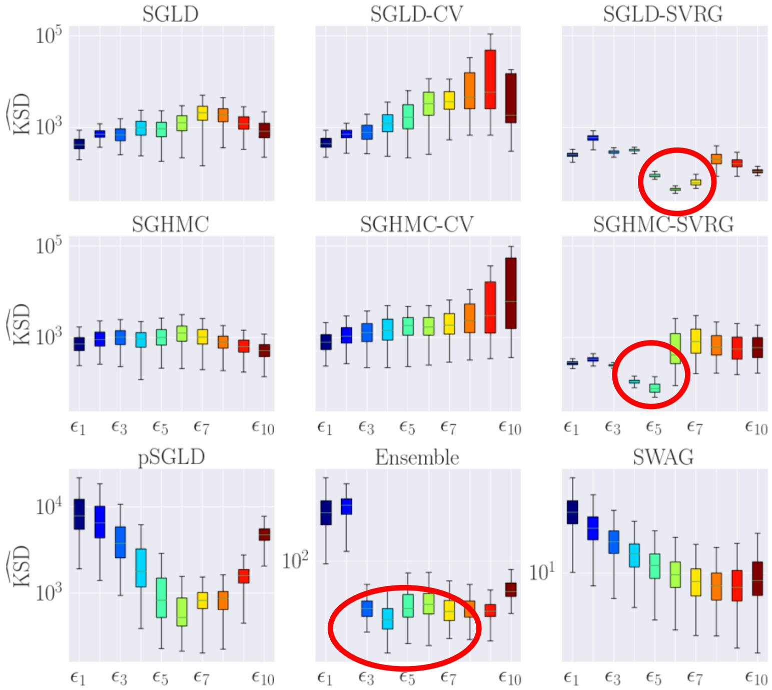
SAFRAN

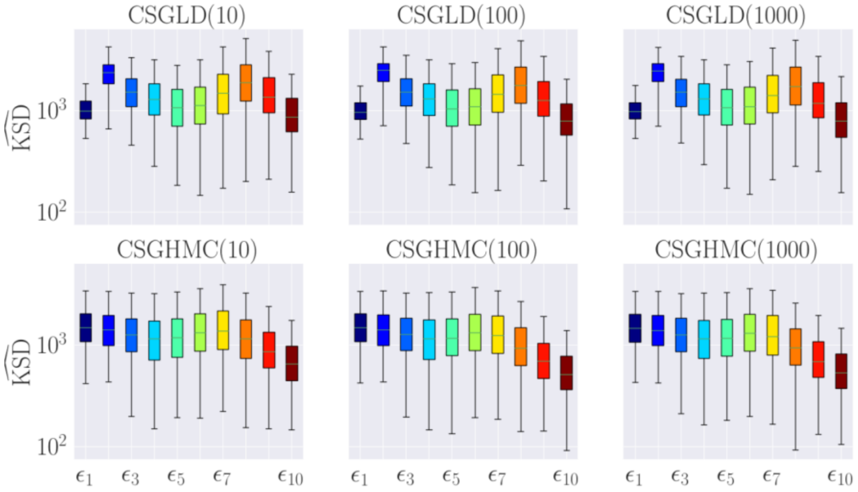
Regression problem #1

KSD distance to the target posterior

Ranking in terms of KSD:

- SWAG < Ensemble < SVRG & pSGLD
- High variability for SGMCMC and C-SGMCMC

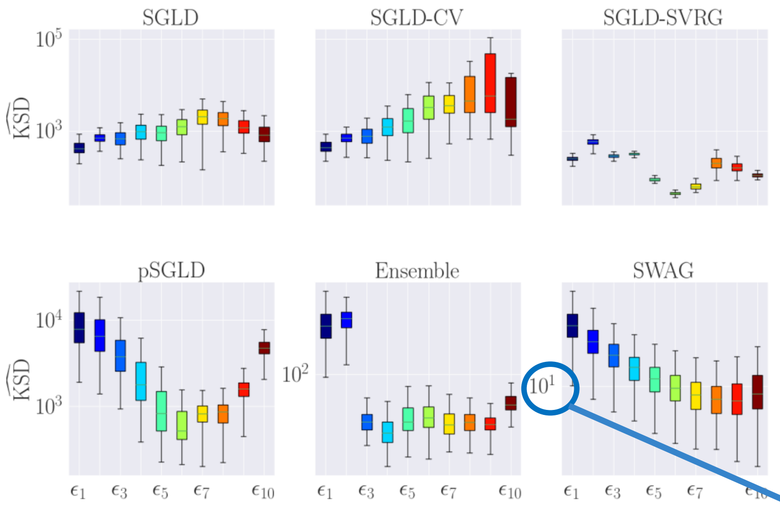


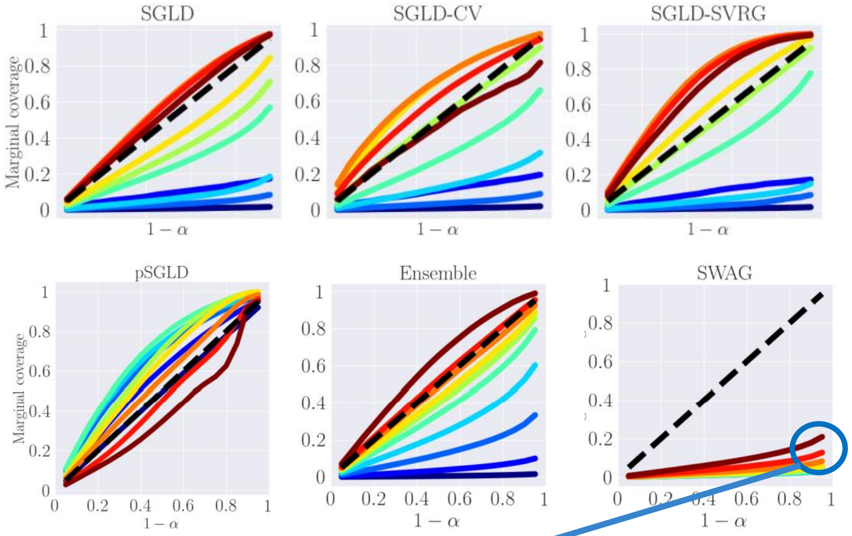


Observed trends in our experiments

- KSD seems uncorrelated with coverage probabilities
- **SWAG** : Lowest KSDs but bad marginal coverages
- **Ensemble** : nice coverages, decent KSDs
- **SVRG** : acceptable coverages, higher KSDs

KSD cannot be exclusively relied on





Pathologies of kernelized Stein discrepancy

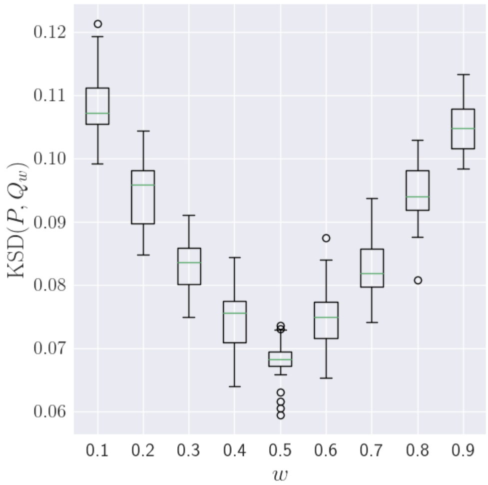
KSD suffers from a few shortcomings [Wenliang et al, 2021; Korba et al, 2021]

› We identified at least two pathologies
› **Pathology I**: blindness to proportions in multimodal distributions

Example

› Target: bimodal Gaussian mixture
› Proportions : 0.25 and 0.75

› Candidates : bimodal Gaussian mixtures with weights w and $1 - w$
› KSD is unable to identity the correct proportions
› KSD seems unreliable when dealing with multimodal distributions



Observed trends in our experiments

Which methods generate valid confidence intervals ?

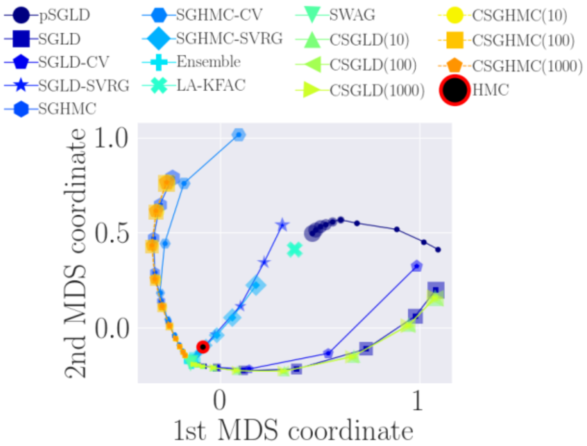
- Several methods are able to provide good marginal/conditional coverage probabilities
- **SGMCMC-SVRG** and **Deep ensembles** seem promising but computationally expensive
- **LA-KFAC** and **pSGLD** easily overshoot the target coverage
- **MC Dropout** or **SWAG** struggle

MMD distances to the HMC reference

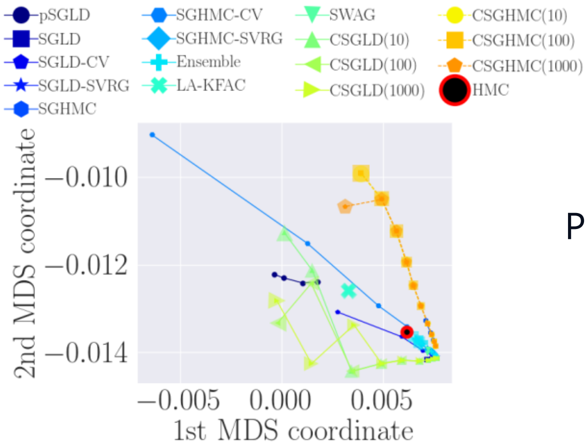
- The behavior w.r.t the step size is not the same in weight and function space
- **SGMCMC-SVRG** and/or **Deep ensembles** usually have the lowest distances in **function space**
- There exist structured similarities in weight space

Same similarities across our experiments

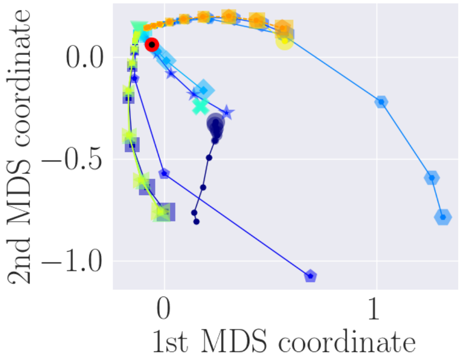
Problem 1



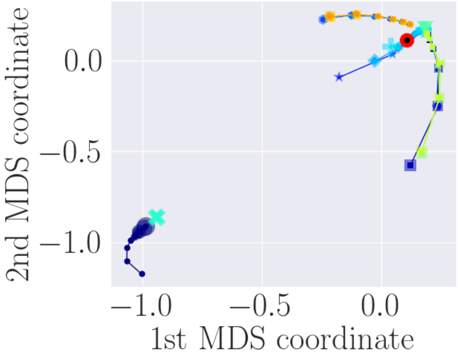
Problem 3



Problem 2



Problem 4



Observed trends in our experiments

KSD distance to the target posterior

- **SWAG** yields low KSD values
- Amongst SGMCMC, **SVRG** variants yield the lowest values
- **Deep ensembles** has slightly lower KSDs than **SVRG variants**

Other comments

- Tuning the hyperparameters is difficult
- KSD should be used with caution, cannot be used for hyperparameter tuning
- Difficult to draw general conclusions from these experiments only

Ongoing / related works

- Running the benchmark with convolutional / graphs networks
- Correction of identified pathologies in the KSD [Benard, Staber, Da Veiga, arxiv:2301.13528]

Implementation details

Laplace approximations in deep learning

- Daxberger et al. (2021): https://github.com/AlexImmer/Laplace
- PyTorch backend

Remaining algorithms

- Implemented with JAX (https://github.com/google/jax)
- Several (SG)MCMC methods are available in BlackJAX (https://github.com/blackjax-devs/blackjax)
- Great speed-up with JAX, especially for MCMC methods (jit, scan, vmap, pmap, etc.)
- Code to reproduce the benchmark will be published soon

Thank you!

References

- Welling, M., & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 681-688).
- Li, C., Chen, C., Carlson, D., & Carin, L. (2016, February). Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).
- Chen, T., Fox, E., & Guestrin, C. (2014, June). Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning* (pp. 1683-1691). PMLR.
- Ma, Y. A., Chen, T., & Fox, E. B. (2015). A complete recipe for stochastic gradient MCMC. *arXiv preprint arXiv:1506.04696*.
- Nemeth, C., & Fearnhead, P. (2020). Stochastic gradient markov chain monte carlo. *Journal of the American Statistical Association*, 1-18.
- Baker, J., Fearnhead, P., Fox, E. B., & Nemeth, C. (2019). Control variates for stochastic gradient MCMC. *Statistics and Computing*, *29*(3), 599-615.
- Dubey, A., Reddi, S. J., Póczos, B., Smola, A. J., Xing, E. P., & Williamson, S. A. (2016). Variance reduction in stochastic gradient Langevin dynamics. *Advances in neural information processing systems*, *29*, 1154.
- Yao, J., Pan, W., Ghosh, S., & Doshi-Velez, F. (2019). Quality of uncertainty quantification for Bayesian neural network inference. *arXiv preprint arXiv:1906.09686*.
- Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., & Hennig, P. (2021). Laplace redux-effortless bayesian deep learning. Advances in Neural Information Processing Systems, 34, 20089-20103.
- Zhang, R., Li, C., Zhang, J., Chen, C., & Wilson, A. G. (2019). Cyclical stochastic gradient MCMC for Bayesian deep learning. arXiv preprint arXiv:1902.03932.
- Gorham, J., & Mackey, L. (2017, July). Measuring sample quality with kernels. In International Conference on Machine Learning (pp. 1292-1301). PMLR.
- Liu, Q., Lee, J., & Jordan, M. (2016, June). A kernelized Stein discrepancy for goodness-of-fit tests. In International conference on machine learning (pp. 276-284). PMLR.
- Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., & Smola, A. (2006). A kernel method for the two-sample-problem. Advances in neural information processing systems, 19.

POWERED BY TRUST

SAFRAN