Multiple Moment Matching Inference: A Flexible Approximate Inference Algorithm

Haonan Duan [1 2] **Pascal Poupart** [1 2]

Abstract

Choosing the right approximation distribution family for variational inference (VI) requires case-by-case design to strike a good balance between representation power and computational cost. Particle-based variational inference (PVI) methods, such as Stein Variational Gradient Descent (SVGD), are proposed as more flexible alternatives to VI due to their non-parametric nature. However, the demand for more flexible inference algorithms never stops. In this paper, we present Multiple Moment Matching Inference (MMMI), a PVI algorithm based on the idea of moment matching. We argue that MMMI allows for more flexible priors and likelihood functions than other PVI algorithms, which further improves its representation power and extends potential application domains. We demonstrate MMMI's competitive predictive performance for Bayesian neural networks on several real-world datasets.

1. Introduction

With recent advances in approximate inference algorithms (Hoffman et al., 2013; Chen et al., 2014), Bayesian methods have proven successful in larger datasets and more complex models (Michelmore et al., 2020). The central problem in Bayesian inference is to approximate the intractable posterior. Variational inference turns this inference problem into deterministic optimization by finding the closest distribution to the posterior in a given approximation set. However, choosing the right approximation distribution family requires case-by-case design to strike a good balance between representation power and computational cost.

Particle-based variational inference (PVI) methods, such as Stein Variational Gradient Descent (SVGD) (Liu & Wang, 2016), are proposed as more flexible alternatives to variational inference, which do not require users to specify an approximation set. SVGD transforms a set of particles to match the posterior distribution in the steepest direction that minimizes the Kullback–Leibler divergence. The non-parametric nature of SVGD allows the posterior to take a more flexible form. Furthermore, eliminating the need of specifying approximation sets eases the application of the method to various domains without ML experts. Thanks to these improvements in flexibility, SVGD has been shown successful in multiple complex tasks (Zhu & Zabaras, 2018).

A natural question to ask is: is it possible to further improve this flexibility? We propose Multiple Moment Matching Inference (MMMI), a PVI algorithm based on the idea of moment matching. We argue that MMMI allows for more flexible priors and likelihoods than other PVI algorithms, which further widens the approximate set for posteriors and also extends its use to black-box models:

- **Flexible priors**: MMMI does not require the prior to be specified in a parametric form. Other methods, such as SVGD, need a parametric prior distribution to compute the unnormalized posterior. Our prior in Bayesian neural networks can be generated directly from well-studied initialization algorithms (Glorot & Bengio, 2010), which extends the potential distribution family of priors and posteriors.

- **Flexible likelihoods**: Different from most approximate inference algorithms like SVGD, our method MMMI does not need gradients of the likelihood function. This can speed up inference where gradient computation is slow and expensive. Furthermore, this means MMMI can be applied to the setting where gradients are undefined (such as discrete objectives) or unknown (such as black-box models).

The paper is organized as follows. Section 2 presents our main algorithm, MMMI. Section 3 reviews the related work in approximate inference and moment matching. Section 4 presents the experimental results of Bayesian neural networks with MMMI on real-world datasets. The paper ends with discussions for future work.

[1] David R. Cheriton School of Computer Science, University of Waterloo, Canada [2] Vector Institute, Ontario, Canada. Correspondence to: Haonan Duan <h4duan@uwaterloo.ca>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning., Copyright 2021 by the author(s).

2. Method

We consider sequential Bayesian inference, where observations are streaming. Training in batches is common in modern deep learning, and Bayesian methods lend themselves naturally to online inference (Broderick et al., 2013).

Our goal is to infer the unknown parameter Θ. For illustration purposes, we assume that Θ is one-dimensional. The algorithm can be easily extended to multi-dimensions. Given a set of particles (samples) $\{\theta_i\}_{i=1}^n$ for the prior distribution $P(\Theta)$, we want to transform them to match the posterior distribution $P(\Theta|X)$ without computing its analytical form. A key observation is that the posterior moments can be estimated by reweighting the samples of the prior distribution, as shown in Subsection 2.1. Then the particles will be transformed in a direction that minimizes the discrepancy between the prior and posterior moments with respect to a given function set, as shown in Subsection 2.2. Afterwards, the transformed particles will represent the prior for the next observation.

2.1. Estimation of Posterior Moments

Given particles $\{\theta_i\}_{i=1}^n$ for the prior distribution $P(\Theta)$, we can estimate a prior moment for the function [1] f_j as a sample average:

$$\mathbb{E}_{\Theta\sim P(\Theta)}[f_j(\Theta)] \approx \sum_{i=1}^n f_j(\theta_i) \tag{1}$$

Directly applying Equation 1 to estimate posterior moments requires sampling from the posterior distributions, which is a difficult task on its own. However, with some simple derivations, we show that posterior moments of f_j can be estimated using only prior samples $\{\theta_i\}_{i=1}^n$.

$$\begin{aligned}\mathbb{E}_{\Theta\sim P(\Theta|X)}[f_j(\Theta)] &= \int_\Theta f_j(\Theta)P(\Theta|X)d\Theta \\ &= \int_\Theta f_j(\Theta)\frac{P(X|\Theta)P(\Theta)}{P(X)}d\theta \\ &= \frac{\int_\Theta f_j(\Theta)P(X|\Theta)P(\Theta)d\Theta}{\int_\Theta P(X|\Theta)P(\Theta)d\Theta} \\ &= \frac{\mathbb{E}_\Theta[f_j(\Theta)P(X|\Theta)]}{\mathbb{E}_\Theta[P(X|\Theta)]} \\ &\approx \frac{\sum_{i=1}^n f_j(\theta_i)P(X|\theta_i)}{\sum_{i=1}^n P(X|\theta_i)}\end{aligned} \tag{2}$$

As shown in Figure 1, our estimation scheme for posterior moments can be interpreted as a weighted average of prior samples, with weights being linearly proportional to the likelihoods. The particles with high likelihood on the observation will contribute more to the estimation of posterior moments.

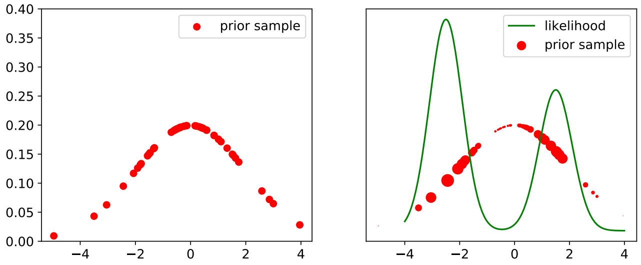

Figure 1. Left figure: When estimating prior moments, we use the unweighted average of prior samples. Right figure: When estimating posterior moments, we use the weighted average of prior samples, where weights are proportional to the likelihood. Note that the $1d$-samples are drawn along their density curve for better visualization.

2.2. Multi-objective Optimization

We use $\hat{\mu}_{f_j}$ to denote the estimation of posterior moments μ_{f_j} from Equation 2, i.e., $\hat{\mu}_{f_j} \approx \mathbb{E}_{\Theta\sim P(\Theta|X)}[f_j(\Theta)]$

As a reminder, our goal is to transform the prior particles $\{\theta_i\}_{i=1}^n$ to the posterior particles $\{\tilde{\theta}_i\}_{i=1}^n$ through moment matching. In other words, we want the moments of transformed particles $\frac{1}{n}\sum_{i=1}^n f_j(\tilde{\theta}_i)$ to match the corresponding target value $\hat{\mu}_{f_j}$. Our objective can be formulated as finding $\{\tilde{\theta}_i\}_{i=1}^n$ to minimize the discrepancy with $\hat{\mu}_{f_j}$:

$$\min_{\tilde{\boldsymbol{\theta}}:=(\tilde{\theta}_1\cdots\tilde{\theta}_n)} S_j(\tilde{\boldsymbol{\theta}}) := \left\|\frac{1}{n}\sum_{i=1}^n f_j(\tilde{\theta}_i) - \hat{\mu}_{f_j}\right\|_2^2 \tag{3}$$

The gradient of the objective in Equation 3 is:

$$\nabla S_j(\tilde{\boldsymbol{\theta}}) = \frac{2}{n}\left(\frac{1}{n}\sum_{i=1}^n f_j(\tilde{\theta}_i) - \hat{\mu}_{f_j}\right)\nabla f_j(\tilde{\boldsymbol{\theta}}) \tag{4}$$

Note that Equation 4 does not require computing the gradients of the likelihood. Computing $\nabla f_j(\tilde{\boldsymbol{\theta}})$ is usually much easier than the gradient of the likelihood. The likelihood function is usually parameterized by complex models, such as neural networks, while f is chosen by the user and usually in a simple form.

In most situations, we want to match more than one moment for different f_j's. Actually, only matching the first moment could lead to particle collapsing. The Hausdorff moment theorem states that the moments of all orders (from 0 to ∞) uniquely determine a distribution defined on a

[1] In this paper, we use a generalized notion of moments. f_j is not restricted to be power functions. We only require f_j to be differentiable.

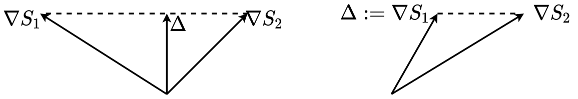

Figure 2. Visualization of the minimum-norm element $\boldsymbol{\Delta}$ in the convex hull U of ∇S_1 and ∇S_2. By inspection, the angles between $\boldsymbol{\Delta}$ and all elements in U are smaller than 90°.

bounded region (Hausdorff, 1921). Therefore, intuitively, the more moments are matched, the better is the approximation. Matching multiple moments is a multi-objective optimization problem, where each objective S_j corresponds to minimizing discrepancy for one moment of function f_j.

Multiple Gradient Descent Algorithm (MGDA) is an iterative gradient-based multi-objective optimization algorithm that updates the parameter in a direction that decreases all objectives simultaneously (Mukai, 1980) (Fliege & Svaiter, 2000)(Désidéri, 2012).

In each iteration, MGDA aims to find a vector $\boldsymbol{\Delta}$ such that:

$$(\boldsymbol{\Delta}, \nabla S_j(\tilde{\boldsymbol{\theta}})) \geq 0, \quad j = 1...m \tag{5}$$

In other words, we want to find a direction $\boldsymbol{\Delta}$ whose angle with each gradient $\nabla S_j(\tilde{\boldsymbol{\theta}})$ is smaller than 90°. This way, following the direction of $-\boldsymbol{\Delta}$, the value of all objective functions $\{S_j(\tilde{\boldsymbol{\theta}})\}_{j=1}^{m}$ will decrease simultaneously.

The problem of finding the direction $\boldsymbol{\Delta}$ is equivalent to finding the min-norm element of the convex hull formed by the gradients $\nabla S_j(\tilde{\boldsymbol{\theta}})$ (Mukai, 1980) (Fliege & Svaiter, 2000)(Désidéri, 2012), which has been studied extensively in computational geometry (Makimoto et al., 1994). Figure 2 shows some intuitions of this idea in $2d$ space. (Sener & Koltun, 2018) proposed an optimization method based on the Frank-Wolfe algorithm to solve the min-norm problem efficiently in high-dimension settings. Following the direction of $\boldsymbol{\Delta}$, MGDA will eventually converge to Pareto-stationarity under some mild conditions.

The pseudocode for MMMI is in Algorithm 1. For each observation, we first estimate posterior moments using Equation 2. Then we compute the gradient for each moment according to Equation 4. In the end, we apply MGDA to find a direction that minimizes the discrepancy for each moment, which is then be applied to update the particles.

3. Related Work

3.1. Particle-based variational inference method

Our algorithm, MMMI belongs to the family of PVI algorithms, which perform deterministic updates on a set of particles to transform them towards the posterior distribution. The motivation of particle transformation comes from bridging the gap between variational inference and MCMC. Like variational inference, those algorithms perform iterative deterministic updates to decrease the distance with the target distribution, while having the advantage of being non-parametric and generic like MCMC.

Algorithm 1 MMMI

```
Input: Particles $\boldsymbol{\theta} := (\theta_1, \cdots \theta_i \cdots, \theta_n)$. Function set $\{f_j\}_{j=1}^{m}$. Data $\{X_k\}_{k=1}^{d}$. Learning rate $\lambda$.
for each data point $X_k$ do
    for each particle $\theta_i$ do
        Obtain the likelihood $L_i$ for $\theta_i$
    end for
    for each function $f_j$ do
        $\hat{\mu}_{f_j} \leftarrow \frac{\sum_i^n L_i f_j(\theta_i)}{\sum_i^n f_j(\theta_i)}$
        $\nabla S_j \leftarrow \frac{2}{n}\left(\frac{1}{n}\sum_i^n f_j(\theta_i) - \hat{\mu}_{f_j}\right) \nabla f_j(\boldsymbol{\theta})$
    end for
    $\boldsymbol{\Delta} \leftarrow MGDA(\nabla S_1, \cdots, \nabla S_m)$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \lambda\boldsymbol{\Delta}$
end for
```

The representative of PVI algorithms is SVGD (Liu & Wang, 2016), which performs functional gradient descent in kernel Hilbert space to minimize KL divergence. The update equation of SVGD for one particle θ_j is: $\hat{\theta}_j \leftarrow \hat{\theta}_j - \epsilon[\sum_{i=1}^{n} k(\theta, \theta_i)\nabla_\theta \log q(\theta_i|D) + \nabla_\theta k(\theta_i, \theta)]$, where $k(\theta, \theta_i)$ is the chosen kernel function. The first term of the update equation can be interpreted as driving the particles into high-likelihood areas, while the second term is pushing the particles away from each other to encourage multi-modality.

There is then lots of follow-up work proposed to improve SVGD from different perspectives. Stein Variational Newton method (SVN) (Detommaso et al., 2018) turns SVGD into a Newton-like iteration in function space by incorporating second-order information. To solve the problem of mode collapse in SVGD, Wang et al. proposed to use particles to represent functions directly instead of samples (Wang et al., 2018), while message passing SVGD aims to reduce the high-dimensional space into a set of local ones over the Markov blanket with lower dimensions (Zhuo et al., 2018).

Compared with MMMI, all of the above algorithms require parametric priors and gradients of the likelihoods.

3.2. Moment Matching

Moments can be seen as a quantitative summary for the shape of a probability distribution. The method of moments is one of the simplest approaches for parameter estimation. It enforces the constraint that population moments (function of unknown parameters) are equal to sample moments (numeric values), turning the problem of parameter estimation

	N	d	BP	SVGD	MMMI
Banana	5300	2	0.847 ± 0.021	0.867 ± 0.015	$\mathbf{0.874 \pm 0.012}$
Diabetis	768	8	0.779 ± 0.044	0.785 ± 0.040	$\mathbf{0.801 \pm 0.042}$
German	1000	20	0.771 ± 0.046	0.778 ± 0.051	$\mathbf{0.793 \pm 0.043}$
Image	2086	18	0.898 ± 0.010	$\mathbf{0.902 \pm 0.013}$	0.899 ± 0.017
Ringnorm	7400	20	0.980 ± 0.003	0.982 ± 0.002	$\mathbf{0.985 \pm 0.003}$
Splice	2991	60	0.916 ± 0.025	$\mathbf{0.926 \pm 0.020}$	0.916 ± 0.026
Two norm	7400	20	0.975 ± 0.009	0.983 ± 0.006	$\mathbf{0.987 \pm 0.005}$
Waveform	5000	21	0.924 ± 0.009	0.929 ± 0.014	$\mathbf{0.941 \pm 0.017}$
Australia Weather	142193	24	0.846 ± 0.004	$\mathbf{0.856 \pm 0.007}$	0.847 ± 0.010
IMDB	50000	100	0.836 ± 0.005	0.839 ± 0.006	$\mathbf{0.841 \pm 0.006}$
Covertype	581012	54	0.812 ± 0.003	0.814 ± 0.007	$\mathbf{0.828 \pm 0.023}$

Table 1. Comparison of frequentist BP, SVGD, MMMI on the 8 classification datasets for neural networks. We report the average and standard deviation of test accuracy over 10 random seeds.

into solving a system of equations

Using differences between moments to measure distance is not new in machine learning. Maximum mean discrepancy has been used as a simple and effective metric in all areas of machine learning, such as generative models (Li et al., 2015) (Sutherland et al., 2017), reinforcement learning (Nguyen et al., 2021) and language embeddings (Yang et al., 2018).

While most studies about moment matching have been frequentist, the idea can lend itself naturally to approximate Bayesian inference. Bayesian moment matching (BMM) projects the mixture posterior to a distribution in the same family as the prior by matching a set of sufficient moments. BMM has been successfully applied in topic modelling (Hsu & Poupart, 2016), sum-product networks (Rashwan et al., 2016) and boolean satisfiablity (Duan et al., 2020). However, different from our algorithm, BMM requires the posterior moments to have a closed-form.

3.3. SVGD as Moment Matching

(Liu & Wang, 2018) shows that SVGD matches the moments of the posterior distribution implicitly. More formally, the fixed-point conditions of the SVGD updates guarantee that the particles $\{\theta_i\}_{i=1}^n$ are transformed to match the expectations of all the functions in a stein matching set F^*. However, unlike our algorithm MMMI, users cannot choose which moment they want to match in SVGD. Choosing a specific moment f in SVGD is equivalent to solving a difficult differential equation with no guarantee of closed-form solutions. Success of SVGD confirms the effectiveness of moment matching in general-purpose approximate inference problems. Our algorithm MMMI further offers a more explicit, more flexible and simpler approach to do moment matching in Bayesian learning.

4. Experiments

We compare our algorithm MMMI with backpropagation (BP) and SVGD on neural networks. We use 8 datasets from the paper of SVGD (Liu & Wang, 2016) plus 3 Kaggle datasets: Australia Weather (Young, 2020), IMDB (Maas et al., 2011), and covertype (Blackard, 1998).

We use a neural network of one hidden layer with 50 units and RELU as the activation function. For Backpropagation, we use the PyTorch (Paszke et al., 2019) implementation with Adam optimizer. For MMMI and SVGD, we use 200 particles for Banana, Diabetis, German plus Image, and 500 for the other datasets. The implementation and the parameters for SVGD are chosen to be the same as the original paper (Liu & Wang, 2016). For MMMI, the prior particles are generated from the default initialization (Glorot & Bengio, 2010) with a small Gaussian perturbation. We match the first and second marginal moments. We find that running MGDA for only 1 iteration is sufficient to obtain good results. We also use Adam (Kingma & Ba, 2014) to update the step size of the gradients.

The inputs of all datasets are standardized to mean 0 and variance 1. All datasets are randomly split according to the ratio $90 : 10$. The missing values are imputed by the mean of the same column. All 3 algorithms perform 10 epochs for the 8 smaller datasets and 1 epoch for the Kaggle datasets. For Australia Weather, we split the date feature into 3 features: year, month and day. For IMDB, we transform customers' reviews to $100-$dimension vectors using Sent2vec (Pagliardini et al., 2018). We find that our algorithm MMMI achieves the best results for most datasets.

5. Conclusion and Future Work

In this paper, we propose MMMI, a simple and flexible particled-based Bayesian inference problem using the idea of moment matching. In the future, we hope to perform a theoretical analysis of MMMI, such as deriving convergence properties and asymptotic behaviors. Also, we plan to apply MMMI on larger models and more interesting applications.

References

Blackard, J. A. *Comparison of neural networks and discriminant analysis in predicting forest cover types*. Colorado State University, 1998.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. Streaming variational bayes. In *NIPS*, 2013.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pp. 1683–1691. PMLR, 2014.

Désidéri, J.-A. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

Detommaso, G., Cui, T., Spantini, A., Marzouk, Y., and Scheichl, R. A stein variational newton method. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 9187–9197, 2018.

Duan, H., Nejati, S., Trimponias, G., Poupart, P., and Ganesh, V. Online bayesian moment matching based sat solver heuristics. In *International Conference on Machine Learning*, pp. 2710–2719. PMLR, 2020.

Fliege, J. and Svaiter, B. F. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Hausdorff, F. Summationsmethoden und momentfolgen. i. *Mathematische Zeitschrift*, 9(1):74–109, 1921.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5), 2013.

Hsu, W.-S. and Poupart, P. Online bayesian moment matching for topic modeling with unknown number of topics. In *NIPS*, pp. 4529–4537, 2016.

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

Li, Y., Swersky, K., and Zemel, R. Generative moment matching networks. In *International Conference on Machine Learning*, pp. 1718–1727. PMLR, 2015.

Liu, Q. and Wang, D. Stein variational gradient descent: a general purpose bayesian inference algorithm. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2378–2386, 2016.

Liu, Q. and Wang, D. Stein variational gradient descent as moment matching. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8868–8877, 2018.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Makimoto, N., Nakagawa, I., and Tamura, A. An efficient algorithm for finding the minimum norm point in the convex hull of a finite point set in the plane. *Operations research letters*, 16(1):33–40, 1994.

Michelmore, R., Wicker, M., Laurenti, L., Cardelli, L., Gal, Y., and Kwiatkowska, M. Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7344–7350. IEEE, 2020.

Mukai, H. Algorithms for multicriterion optimization. *IEEE Transactions on Automatic Control*, 25(2):177–186, 1980.

Nguyen, T. T., Gupta, S., and Venkatesh, S. Distributional reinforcement learning via moment matching. *AAAI*, 2021.

Pagliardini, M., Gupta, P., and Jaggi, M. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Rashwan, A., Zhao, H., and Poupart, P. Online and distributed bayesian moment matching for parameter learning in sum-product networks. In *Artificial Intelligence and Statistics*, pp. 1469–1477. PMLR, 2016.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 525–536, 2018.

Sutherland, D. J., Tung, H.-Y., Strathmann, H., De, S., Ramdas, A., Smola, A. J., and Gretton, A. Generative models and model criticism via optimized maximum mean discrepancy. In *ICLR (Poster)*, 2017.

Wang, Z., Ren, T., Zhu, J., and Zhang, B. Function space particle optimization for bayesian neural networks. In *International Conference on Learning Representations*, 2018.

Yang, P., Luo, F., Wu, S., Xu, J., Zhang, D., and Sun, X. Learning unsupervised word mapping by maximizing mean discrepancy. *arXiv preprint arXiv:1811.00275*, 2018.

Young, J. Rain in australia, 2020.

Zhu, Y. and Zabaras, N. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366:415–447, 2018.

Zhuo, J., Liu, C., Shi, J., Zhu, J., Chen, N., and Zhang, B. Message passing stein variational gradient descent. In *International Conference on Machine Learning*, pp. 6018–6027. PMLR, 2018.