# Supplementary material

## A Derivation of a counterexample in Sec. 3

Our derivations follow (Girard et al., 2003), who study GPs with uncertain inputs. Specifically they compute the mean and the variance of $f(x_*)$, where $f \sim \mathcal{GP}(\mu(\cdot), k(\cdot,\cdot))$, and $x_* \sim \mathcal{N}(\mu_*, \sigma_*^2)$.

According to Eq. (12) in Girard et al. (2003),

$$\begin{aligned} v(\mu_*, \sigma_*^2) :&= \mathbb{V}\text{ar}\left[f(x_*)\right] \\ &= 1 + \operatorname{Tr}\left[(\beta\beta^T - \mathbf{K}^{-1})Q\right] - \operatorname{Tr}(\mathbf{q}^T\beta)^2, \end{aligned} \tag{25}$$

where $\mathbf{K}_{ij} = k(\mathbf{z}_i, \mathbf{z}_j)$ is a kernel matrix at inducing locations. Assuming for simplicity a squared-exponential kernel $k(x, x') = \exp((x - x')^2/2\gamma^2)$, and a single inducing point $(z, u)$, $\mathbf{K}$ becomes a scalar, $\mathbf{K} = 1$. The matrix $Q$ in the equation above has as many rows and columns as there are inducing points, meaning that under our assumptions, $Q$ is a scalar given by the following equation:

$$Q = \frac{1}{\sqrt{\frac{2\sigma_*^2}{\gamma^2} + 1}} \exp\left(-\frac{1}{2\left(\frac{\gamma^2}{2} + \sigma_*^2\right)}(u - \mu_*)^2\right).$$

The term $\beta$ in (25) is defined as $\beta = \mathbf{K}^{-1}\mathbf{u}$, which equals zero assuming that the single inducing point $u$ is equal to zero.

In summary, under our assumptions,

$$v(\mu_*, \sigma_*^2) := \mathbb{V}\text{ar}\left[f(x_*)\right] = 1 - Q. \tag{26}$$

The derivative of (26) w.r.t. $\sigma_*^2$ is as follows

$$\begin{aligned} \frac{\partial v(\mu_*, \sigma_*^2)}{\partial \sigma_*^2} &= \exp\left(-\frac{1}{2\left(\frac{\gamma^2}{2} + \sigma_*^2\right)}(u - \mu_*)^2\right) \times \\ &\times \left(-\frac{(u - \mu_*)^2}{2\left(\frac{\gamma^2}{2} + \sigma_*^2\right)^2} + \frac{1}{\gamma^2\left(\frac{2\sigma_*^2}{\gamma^2} + 1\right)^{3/2}}\right). \end{aligned}$$

Evaluating this derivative at $\sigma_*^2 = 0$, we obtain

$$\begin{aligned} \left.\frac{\partial v(\mu_*, \sigma_*^2)}{\partial \sigma_*^2}\right|_{\sigma_*^2=0} &= \exp\left(-\frac{(u - \mu_*)^2}{\gamma^2}\right) \times \\ &\times \left(-\frac{2(u - \mu_*)^2}{\gamma^4} + \frac{1}{\gamma^2}\right). \end{aligned}$$

From the above equation it is easy to see that

$$\left.\frac{\partial v(\mu_*, \sigma_*^2)}{\partial \sigma_*^2}\right|_{\sigma_*^2=0} < 0 \quad \Leftrightarrow \quad \gamma < \sqrt{2}|u - \mu_*|.$$

In other words, if the input mean is sufficiently far away from the inducing point (in relation to the length scale), *i.e.* $\gamma < \sqrt{2}|u - \mu_*|$, adding input noise may reduce the output uncertainty.

## B Analytic marginalisation of jointly Gaussian inducing points

In this section, we provide the derivation of the variational distribution with analytically marginalised inducing points that have a joint Gaussian distribution, as described in Sec. 4.1. We first derive the result for a 2-layer case and then discuss a way to generalise beyond two layers.

We consider jointly Gaussian inducing points as

$$q(\mathbf{u}_1, \mathbf{u}_2) \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{m}_1 \\ \mathbf{m}_2\end{pmatrix}, \begin{pmatrix}S_{11} & S_{12} \\ S_{21} & S_{22}\end{pmatrix}\right), \tag{27}$$

and a joint variational distribution of a two-layer DGP (suppressing the dependence on the inducing locations $\mathbf{z}_1$ in the notation) is

$$\begin{aligned} q(\mathbf{f}_2, \mathbf{u}_2, \mathbf{f}_1, \mathbf{u}_1) = p(\mathbf{f}_2 \mid \mathbf{u}_2, \mathbf{f}_1) q(\mathbf{u}_2 \mid \mathbf{u}_1) \\ p(\mathbf{f}_1 \mid \mathbf{u}_1, \mathbf{x}) q(\mathbf{u}_1). \end{aligned} \tag{28}$$

The goal is to integrate $\mathbf{u}_1$ and $\mathbf{u}_2$ out from (28) in order to fit the model without sampling the inducing points. The following derivations are based on the argument that the mean of $F_i$ is a linear transformation of $U_i$, and vice versa.

Assume that $q(\mathbf{u}_1) \sim \mathcal{N}(\mathbf{m}_1, S_{11})$ and $p(\mathbf{f}_1 \mid \mathbf{x}) \sim \mathcal{N}(\tilde{\mu}_1, \tilde{\Sigma}_1)$ with

$$\begin{aligned} \tilde{\mu}_1 &= \mu_1(\mathbf{x}) + \alpha_1(\mathbf{x})^T(\mathbf{m}_1 - \mu_1(\mathbf{z}_0)), \\ \tilde{\Sigma}_1 &= K_1(\mathbf{x}, \mathbf{x}) - \alpha_1(\mathbf{x})^T(K_1(\mathbf{z}_0, \mathbf{z}_0) - S_{11})\,\alpha_1(\mathbf{x}), \end{aligned} \tag{29}$$

where $\alpha_1(\mathbf{x}) = K_1(\mathbf{z}_0, \mathbf{z}_0)^{-1}K_1(\mathbf{z}_0, \mathbf{x})$. We can compute the joint distribution $q(\mathbf{u}_1, \mathbf{f}_1 \mid \mathbf{x}) = q(\mathbf{u}_1)p(\mathbf{f}_1 \mid \mathbf{u}_1, \mathbf{x})$ using a standard result[6] for a linear model with a Gaussian prior and likelihood (in the following we will be referring to this result as $(*)$) as follows:

$$q(\mathbf{u}_1, \mathbf{f}_1) \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{m}_1 \\ \tilde{\mu}_1\end{pmatrix}, \begin{pmatrix}S_{11} & S_{11}\alpha_1(\mathbf{x}) \\ \alpha_1(\mathbf{x})^T S_{11} & \tilde{\Sigma}_1\end{pmatrix}\right). \tag{30}$$

From this we can swap $\mathbf{u}_1$ and $\mathbf{f}_1$ in the conditional dis-

[6] See, for example, Section 4 in https://davidrosenberg.github.io/mlcourse/in-prep/multivariate-gaussian.pdf.

tribution by computing

$$q(\mathbf{u}_1 \mid \mathbf{f}_1) \sim \mathcal{N}(\mathbf{m}_1 + S_{11}\alpha_1(\mathbf{x})\tilde{\Sigma}_1^{-1}(\mathbf{f}_1 - \tilde{\mu}_1), \\ S_{11} - S_{11}\alpha_1(\mathbf{x})\tilde{\Sigma}_1^{-1}\alpha_1(\mathbf{x})^T S_{11}). \tag{31}$$

Now we can integrate $\mathbf{u}_1$ from (28) by applying (*) again, obtaining

$$q(\mathbf{f}_2, \mathbf{u}_2, \mathbf{f}_1) = p(\mathbf{f}_2 \mid \mathbf{u}_2, \mathbf{f}_1)q(\mathbf{u}_2 \mid \mathbf{f}_1)p(\mathbf{f}_1 \mid \mathbf{x}), \tag{32}$$

where (33)

$$q(\mathbf{u}_2 \mid \mathbf{f}_1) = \mathcal{N}(\mathbf{m}_2 + S_{21}\alpha_1(\mathbf{x})\tilde{\Sigma}_1^{-1}(\mathbf{f}_1 - \tilde{\mu}_1), \\ S_{22} - S_{21}\alpha_1(\mathbf{x})\tilde{\Sigma}_1^{-1}\alpha_1(\mathbf{x})^T S_{12}).$$

Another application of (*) allows us to integrate $\mathbf{u}_2$ from (33) obtaining joint distribution of intermediate layers $q(\mathbf{f}_2, \mathbf{f}_1 \mid \mathbf{x}) = q(\mathbf{f}_2 \mid \mathbf{f}_1)q(\mathbf{f}_1 \mid \mathbf{x})$ with $q(\mathbf{f}_2 \mid \mathbf{f}_1) = \mathcal{N}(\tilde{\mu}_2, \tilde{\Sigma}_2)$ where

$$\begin{aligned}\tilde{\mu}_2 &= \mu_2(\mathbf{f}_2) + \alpha_2(\mathbf{f}_1)^T(\mathbf{m}_2 + S_{21}\alpha_1(\mathbf{x})\tilde{\Sigma}_1^{-1} \\ &\quad (\mathbf{f}_1 - \tilde{\mu}_1 - \alpha_1(\mathbf{x})^T(\mathbf{m}_1 - \mu_1(\mathbf{z}_0)) - \mu_2(\mathbf{z}_1)), \\ \tilde{\Sigma}_2 &= K_2(\mathbf{f}_1, \mathbf{f}_1) - \alpha_2(\mathbf{f}_1)^T(K_2(\mathbf{z}_1, \mathbf{z}_1) - S_{22} + \\ &\quad + S_{21}\alpha_1(\mathbf{x})\tilde{\Sigma}_1^{-1}\alpha_1(\mathbf{x})^T S_{12})\alpha_2(\mathbf{f}_1).\end{aligned} \tag{34}$$

This result can be generalised to more than two layers, starting with $q(\mathbf{f}_i|\mathbf{f}_{i-1}, ..., \mathbf{f}_1) \sim N(\mu_i, \Sigma_i)$, and repeating the steps outlined above to arrive at $q(\mathbf{f}_{i+1}|\mathbf{f}_i, \mathbf{f}_{i-1}, ..., \mathbf{f}_1)$, parameters of which can be deduced by replacing the indices for the first layer with indices for the i$^{\text{th}}$ layer in (34). This leads to the result given in (16) and (17).

# C Implementation

Our implementations for the approaches discussed in Sec. 4.1 and Sec. 4.2 are built on the Tensorflow (Abadi et al., 2015) and the Tensorflow Probability (Dillon et al., 2017) libraries.

# D Alignment task

Another example of a task that calls for an explicit representation of the constituent functions is the task of aligning temporal sequences (Kaiser et al., 2018; Kazlauskaite et al., 2019). Consider a set of sequences $\{\mathbf{y}_j\}_{j=1}^J$ where each sequence $\mathbf{y}_j \in \mathbb{R}^N$ is observed at fixed inputs $\mathbf{x} \in \mathbb{R}^N$ that typically correspond to time. It is known that the observed sequences were generated by temporally warping the inputs $\mathbf{x}$ as follows:

$$\mathbf{y}_j = f_j(g_j(\mathbf{x})) + \epsilon_j \tag{35}$$

where $g_j(\cdot)$ is the temporal warping, $f_j(\cdot)$ is the latent function that encodes the structure of the observed sequence (that is not corrupted by the temporal warping) and $\epsilon_j \sim \mathcal{N}(0, \sigma_j^2)$ is the observation noise. Given this construction, prior knowledge may be imposed on the two functions that make up the model; for example, the temporal warps are typically constrained to be monotonic increasing to ensure that the order of observations is preserved, while the latent functions may be described using a Gaussian process prior with an appropriate kernel that represent our beliefs about the features of these functions. The goal in an alignment task is to learn the model of the data as defined in (35) such that the latent functions $\{f_j\}$ for all $J$ sequences are as similar as possible, *i.e.* we are interested in such a composition of the functions $f_j$ and $g_j$ such that $\sum_{i=1}^J \sum_{k=i+1}^J (f_i(\mathbf{x}) - f_k(\mathbf{x}))$ (the pairwise distance between the latent functions) is as small as possible given the prior assumptions on $\{f_j\}$ and $\{g_j\}$. The composition in (35) can be expressed using a two-layer DGP with appropriate priors (for a detailed description of imposing monotonicity constraints, see (Ustyuzhaninov et al., 2020)).

Consider a set of 3 sequences generated using a sinc function in the range $[-1, 1]$ that need to be aligned. Fig. D1 illustrates how correlations between layers allow us to uncover a set of solutions, as opposed to a point estimate of the warping and the latent functions reported in (Kazlauskaite et al., 2019).

Some additional correlations need to be introduced into the alignment model to ensure that any given sample of the 3 latent functions $f_j(\mathbf{x}), j = 1, 2, 3$ at fixed inputs $\mathbf{x}$ are consistent (otherwise, the solution collapses to a single latent function for all sequences which is at odds with our goal of finding a range of possible solutions). In this example, the additional correlations are introduced by jointly sampling the inducing points that define the first layer of the composition.

# E Additional numerical simulations

In this section we provide additional examples (Fig. E1 to E6) of 3-layer DGP fits to two functions, a sine and an identity function. Similar to Fig. 5, we fit a DGP to both functions using three variational inference schemes based on a factorised variational distribution of inducing points (DSVI), jointly Gaussian inducing points of Sec. 4.1, and the distribution discussed in Sec. 4.2.

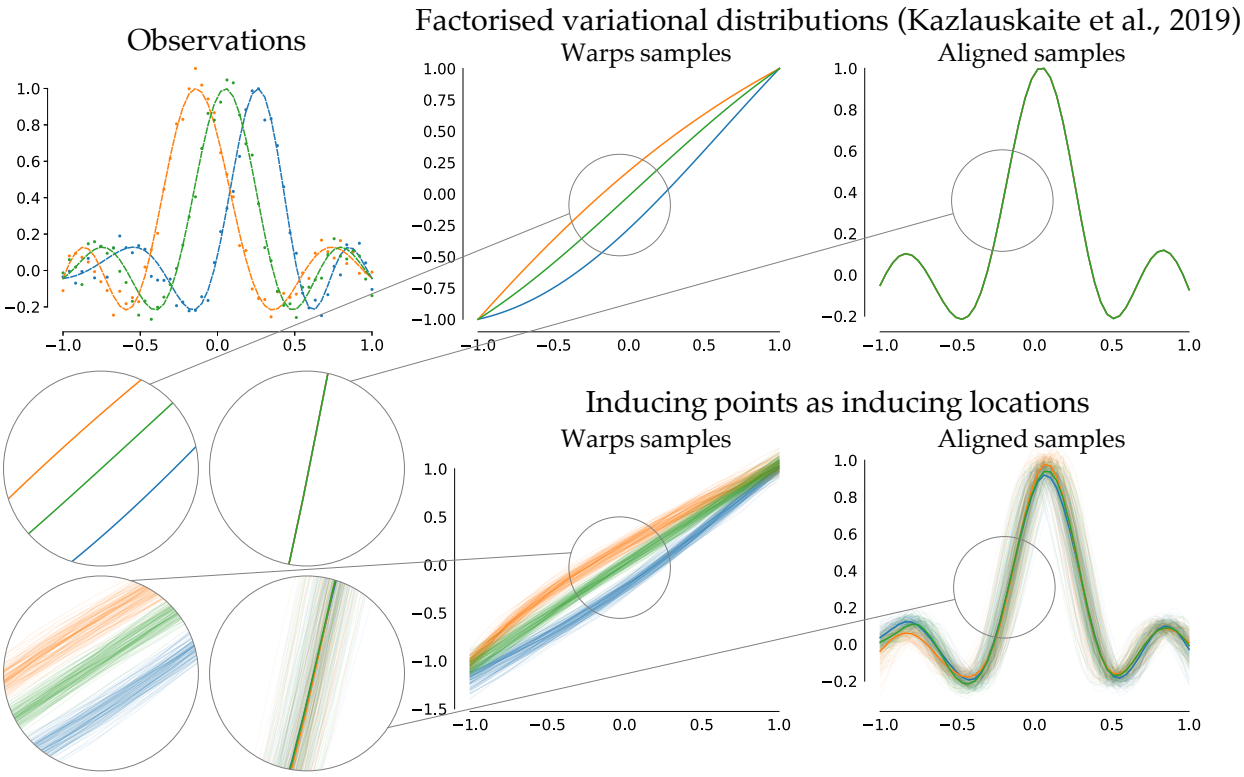


Figure D1: Alignment task. The top left figure shows the observed data that needs to be aligned. The two rows on the right show the alignment used in Kazlauskaite et al. (2019), that provides a point estimate of the solution (top row), and the alignment using a probabilistic model with correlated warping functions and latent functions (bottom row).

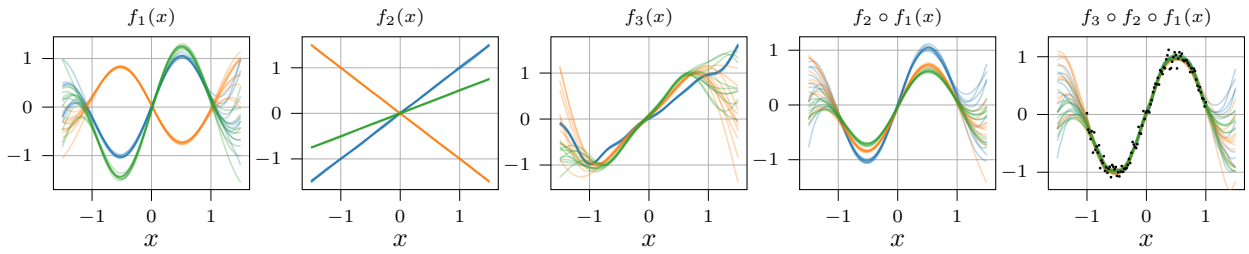


Figure E1: Example fits of a three-layer DGP *with factorised inducing points* to a data set shown in the rightmost panel (black dots). Different panels show the computations performed by each of the three layers and their compositions. Different colours correspond to three models fitted to the same data with different random initialisations. For each initialisation, ten samples (of the same colour) from the fitted model are shown on top of each other.

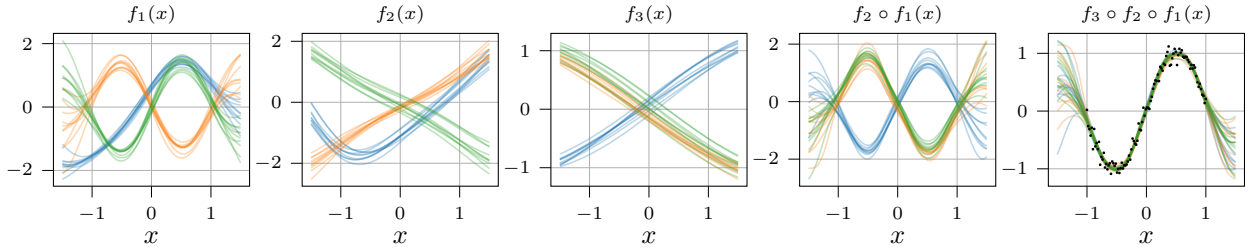


Figure E2: Example fits of a three-layer DGP with *jointly Gaussian inducing points* (Sec. 4.1). The figure arrangement is the same as in Fig. E1.

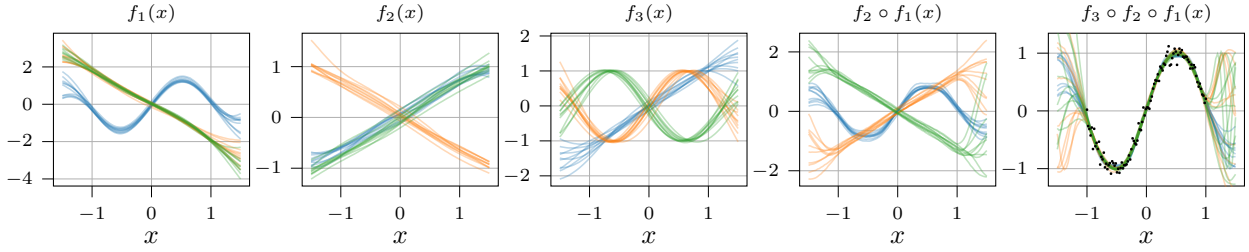


Figure E3: Example fits of a three-layer DGP with *inducing points as inducing locations* (Sec. 4.2). The figure arrangement is the same as in Fig. E1

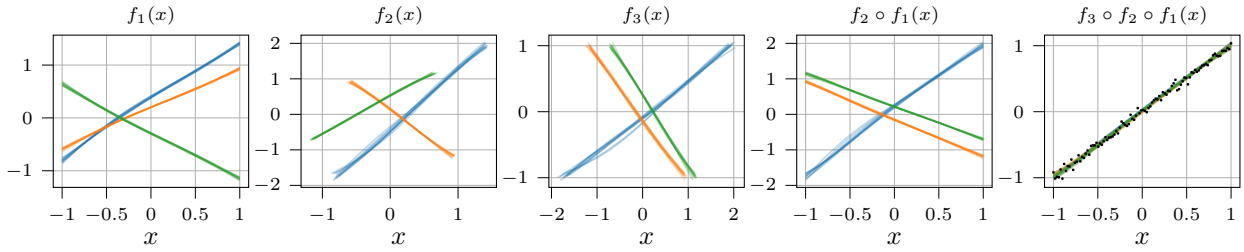


Figure E4: Example fits of a three-layer DGP *with factorised inducing points* to a data set shown in the rightmost panel (black dots). Different panels show the computations performed by each of the three layers and their compositions. Different colours correspond to three models fitted to the same data with different random initialisations. For each initialisation, ten samples (of the same colour) from the fitted model are shown on top of each other.

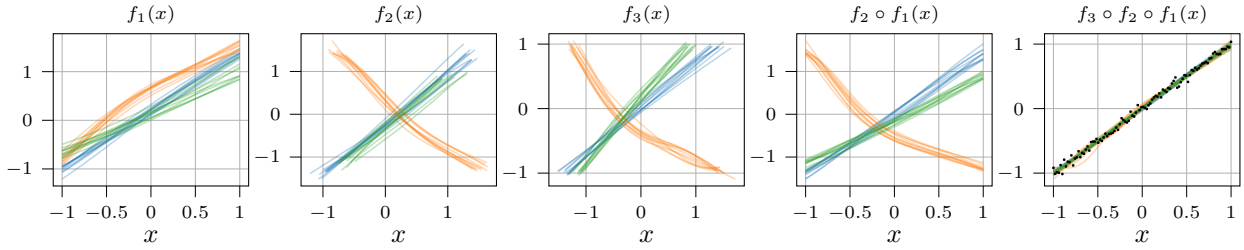


Figure E5: Example fits of a three-layer DGP with *jointly Gaussian inducing points* (Sec. 4.1). The figure arrangement is the same as in Fig. E4.

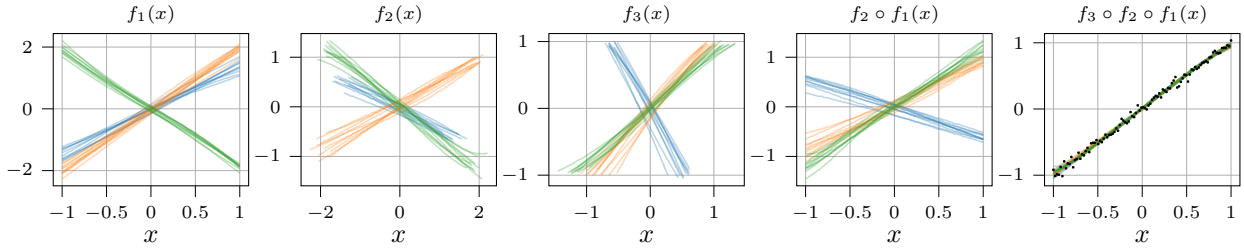


Figure E6: Example fits of a three-layer DGP with *inducing points as inducing locations* (Sec. 4.2). The figure arrangement is the same as in Fig. E4

## References (for appendix)


M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Zh. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, Sh. Moore, D. Murray, Ch. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng. (2015) *TensorFlow: Large-scale machine learning on heterogeneous systems.* Software available from *tensorflow.org.*

J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, R. A. Saurous (2017). *TensorFlow Distributions. arXiv preprint arXiv:1711.10604*