Algorithm 1: Training Procedure.

Input: Forward model f, prior over physical parameters $\mathrm{P}(\boldsymbol{\theta})$.

Output: Approximate posterior $\mathrm{Q}_\phi(\boldsymbol{\theta}|\mathbf{x})$. Also a likelihood $\mathrm{P}_{\mathbf{W}}(\mathbf{x}|\boldsymbol{\theta})$.

1 **repeat**

2 Simulate $\{(\boldsymbol{\theta}_i, \mathbf{x}_i)\}_{i=1}^N$ pairs, using $\mathbf{x}_i \leftarrow f(\boldsymbol{\theta}_i)$, $\boldsymbol{\theta}_i \sim \mathrm{P}(\boldsymbol{\theta}_i)$.

3 Train $\mathrm{Q}_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\mathbf{x})$ via ML:

$$\arg\max_{\boldsymbol{\phi}} \sum_{i=1}^{N} \log \mathrm{Q}_{\boldsymbol{\phi}}(\boldsymbol{\theta}_i|\mathbf{x}_i)$$

4 Train a neural likelihood (or likelihood-ratio?)

$$\arg\max_{\mathbf{W}} \sum_{i=1}^{N} \log \mathrm{P}_{\mathbf{w}}(\mathbf{x}_i|\boldsymbol{\theta}_i)$$

5 Minimise a divergence (e.g. D_{KL}):

$$\arg\min_{\boldsymbol{\phi}} D_{\mathrm{KL}}\big[\mathrm{Q}_\phi(\boldsymbol{\theta}|\mathbf{x}_{\mathrm{true}})\|\mathrm{P}(\boldsymbol{\theta}|\mathbf{x}_{\mathrm{true}})\big]$$

where $\mathrm{P}(\boldsymbol{\theta}|\mathbf{x}_{\mathrm{true}}) \propto \mathrm{P}_{\mathbf{W}}(\mathbf{x}_{\mathrm{true}}|\boldsymbol{\theta})\mathrm{P}(\boldsymbol{\theta})$.

6 **until** *Until reconstructions match the data*

The objective on line 5 has the following form:

$$\begin{aligned}
\mathcal{L}(\phi) &= \mathbb{E}_{\mathrm{Q}_\phi(\boldsymbol{\theta}|\mathbf{x}_{\mathrm{true}})}\big[\log \mathrm{P}(\boldsymbol{\theta}, \mathbf{x}_{\mathrm{true}}) - \log \mathrm{Q}_\phi(\boldsymbol{\theta}|\mathbf{x}_{\mathrm{true}})\big] && (1)\\
&= \mathbb{E}_{\mathrm{Q}_\phi(\boldsymbol{\theta}|\mathbf{x}_{\mathrm{true}})}\big[\log \mathrm{P}_{\mathbf{W}}(\mathbf{x}_{\mathrm{true}}|\boldsymbol{\theta})\big] - D_{\mathrm{KL}}\left[\mathrm{Q}_\phi(\boldsymbol{\theta}|\mathbf{x}_{\mathrm{true}})\|\mathrm{P}(\boldsymbol{\theta})\right] && (2)
\end{aligned}$$

We must be careful however, since each dimension of $Q_\phi(\boldsymbol{\theta}|\mathbf{x})$ (a Sequential Autoregressive Network) is a mixture distribution. In order to compute $\nabla_\phi \mathcal{L}(\phi)$, we must sample from the distribution in such a way that it can be reparametrised; by finding the expectation under each component individually, and then weighting these terms by the mixture weights. That is, for a mixture distribution $Q_\phi(\boldsymbol{\theta}|\mathbf{x}) = \sum_{k=1}^{K} \varphi_\phi^k(\mathbf{x}) Q_\phi^k(\boldsymbol{\theta}|\mathbf{x})$, we can pull the mixture weights out of the expectation:

$$\begin{aligned}
\mathbb{E}_{Q_\phi(\boldsymbol{\theta}|\mathbf{x})} f(\boldsymbol{\theta}) &= \int_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \left(\sum_{k=1}^{K} \varphi_\phi^k(\mathbf{x}) Q_\phi^k(\boldsymbol{\theta}|\mathbf{x})\right) d\boldsymbol{\theta} && (3)\\
&= \sum_{i=1}^{K} \varphi_\phi^k(\mathbf{x}) \int_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) Q_\phi^k(\boldsymbol{\theta}|\mathbf{x}) d\boldsymbol{\theta} && (4)\\
&= \sum_{k=1}^{K} \varphi_\phi^k(\mathbf{x}) \mathbb{E}_{Q_\phi^k(\boldsymbol{\theta}|\mathbf{x})}\big[f(\boldsymbol{\theta})\big]. && (5)
\end{aligned}$$

However, by simply substituting $f(\boldsymbol{\theta}) = \log \mathrm{P}(\boldsymbol{\theta}, \mathbf{x}) - Q_\phi(\boldsymbol{\theta}|\mathbf{x})$ and optimising $\mathcal{L}(\phi)$, the mode-seeking behaviour of the KL-divergence will down-weight many components in the mixture. Roeder et al. (2017) propose to use importance sampling (an IWAE), where we first draw T iid. samples from the posterior. Combining the approach above leads to a "*stratified* IWAE", which is computed as follows:

$$\mathcal{L}^T(\phi) = \mathbb{E}_{\{\boldsymbol{\theta}_{kt} \sim Q_\phi^k(\boldsymbol{\theta}|\mathbf{x}_{\mathrm{true},j})\}_{k=1,t=1}^{K,T}} \left[\log \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{K} \varphi_\phi^k(\mathbf{x}_{\mathrm{true},j}) \frac{\mathrm{P}_{\mathbf{W}}(\boldsymbol{\theta}_{kt}, \mathbf{x}_{\mathrm{true},j})}{Q_\phi(\boldsymbol{\theta}_{kt}|\mathbf{x}_{\mathrm{true},j})}\right], \quad (6)$$

where the notation $\{\boldsymbol{\theta}_{kt} \sim Q_\phi^k(\boldsymbol{\theta}|\mathbf{x})\}_{k=1,t=1}^{K,T}$ means, draw T samples from each of the K mixture components, and $\mathbf{x}_{\mathrm{true},j}$ is the jth observation from the catalogue of real observations.

References

D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. Mastering Atari with Discrete World Models. *arXiv:2010.02193 [cs, stat]*, Dec. 2020. URL `http://arxiv.org/abs/2010.02193`.

G. Roeder, Y. Wu, and D. K. Duvenaud. Sticking the Landing: Simple, Lower-Variance Gradient Estimators for Variational Inference. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/e91068fff3d7fa1594dfdf3b4308433a-Abstract.html`.