1. ADD as Multiplicative Noise and Equivalence to Stochastic Depth

Just as we did for ARD, we can re-formulate ADD in its expanded parametrization, revealing its equivalent MN form:

\[ h_{n,l} = f_l(h_{n,l-1}W_l) + h_{n,l-1} \]

where \( \tau_l \) is the ADD scale random variable. Comparing the equation above to the corresponding equation for ARD, we see that ADD is much less sampling intensive, requiring just one noise variable for each hidden layer.

Huang et al. (2016) proposed stochastic depth resnets by applying dropout to the resnet block, i.e. \( a_l = \lambda_l F_l(h_{l-1}) + h_{l-1} \) where \( F(\cdot) \) denotes a whole resnet block and \( \lambda_l \) is a Bernoulli random variable. For simplicity, if we assume the resnet block consists of just one non-linear transformation, then applying Bernoulli ADD in MN form to the resnet architecture yields a similar expression, as seen in Equation 1. The only difference is that the Bernoulli variable is within \( f_l \) for ADD whereas it is outside the block for stochastic depth. However, if we assume \( f_l \) is the ReLU function, then the activation is scale equivariant. If the noise’s support is non-negative, then it is equivalent to move the noise variable outside the activation: \( \text{ReLU}(\tau_l h_{l-1}W_l) + h_{l-1} = \tau_l \text{ReLU}(h_{l-1}W_l) + h_{l-1} \) for \( \tau \geq 0 \). Thus, we can derive the previously proposed stochastic depth regularization as a special case of our ADD framework.

2. Parametrizations and their Effect on the Jensen’s Gap

Given the equivalence between training under multiplicative noise and MC integration of the scale prior, we wonder: are we working with the correct parametrization? A first glance suggests that sampling noise in the hierarchical parametrization, whereas it is outside the block for stochastic depth. However, if we assume \( f_l \) is a whole resnet block and \( \lambda_l \) is a Bernoulli random variable. For simplicity, if we assume the resnet block consists of just one non-linear transformation, then applying Bernoulli ADD in MN form to the resnet architecture yields a similar expression, as seen in Equation 1. The only difference is that the Bernoulli variable is within \( f_l \) for ADD whereas it is outside the block for stochastic depth. However, if we assume \( f_l \) is the ReLU function, then the activation is scale equivariant. If the noise’s support is non-negative, then it is equivalent to move the noise variable outside the activation: \( \text{ReLU}(\tau_l h_{l-1}W_l) + h_{l-1} = \tau_l \text{ReLU}(h_{l-1}W_l) + h_{l-1} \) for \( \tau \geq 0 \). Thus, we can derive the previously proposed stochastic depth regularization as a special case of our ADD framework.

We first consider the model in its expanded parametrization. Denote the Jensen’s gap in this setting as \( \mathcal{J}_{\text{EP-GAP}} \). The following proposition reveals the gap’s series representation.

**Proposition 2.1.** Let the NN that parametrizes the likelihood be a 2-Lipschitz function, let \( k \) denote an integer, let \( \mu_\xi = \mathbb{E}[\xi] \), and let \( \text{Var}[\xi] \) denote the variance of the scale (noise) distribution. The Jensen’s gap of the expanded parametrization is:

\[
\mathcal{J}_{\text{EP-GAP}} = \frac{1}{2} \text{Var}[\xi] \left\| \nabla_{\mu_\xi} \log p(y|X, \{W_l\}_{l=1}^{L+1}, \{E[\Xi_l]\}_{l=1}^{L}) \right\|_2^2 - \sum_{k=2}^{\infty} \frac{(-1)^k}{k} \left[ \frac{1}{2} \text{Var}[\xi] \right]^k \Omega^k - \frac{1}{8} \text{Var}^2[\xi] \Omega^2 + O(\text{Var}^3[\xi])
\]

---

1Department of Engineering, University of Cambridge, Cambridge, United Kingdom
2Microsoft Research, Cambridge, United Kingdom
3Alan Turing Institute
4Department of Computer Science, University of California, Irvine, United States of America. Correspondence to: Eric Nalisnick <e.nalisnick@eng.cam.ac.uk>.

Pre-print. Work in progress.
The derivation is provided in the supplementary materials. We see that the gap depends on the variance of $\xi$.

The proof is given in the supplementary materials. We see that the gap is a function of the variance of the scale and the norm of the log likelihood’s gradient w.r.t. the scales. This latter quantity could be large. However, as $E[\xi] \rightarrow 0$ so does $\text{Var}[\xi] \rightarrow 0$. We plot $E[\xi^2]$ vs $\text{Var}[\xi]$ in Figure 1 (solid lines) for the noise distributions given in Table 1. Not only is $\text{Var}[\xi]$ near zero when the prior is encouraging strong shrinkage but we also see favorable scaling for all but the inverse Nakagami as $E[\xi^2]$ grows.

We now move on to the hierarchical parametrization in which the random scale is left in the Gaussian prior. Denote the Jensen’s gap as $J_{\text{HP-GAP}}$, and we give its series representation below.

**Proposition 2.2.** Let $p(\mathbf{W}|\Xi^{-2})$ be a factorized, zero-mean Gaussian prior with $\xi^{-2}$ denoting its precision. The Jensen’s gap in the hierarchical parametrization is:

$$J_{\text{HP-GAP}} = \frac{L}{2} \text{Var}[\xi^{-2}] \left\| \mathbf{\nabla}_{\mathbf{\mu}_c} \log p(\mathbf{y}|\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^{L+1}, \{E[\Xi]\}_{i=1}^{L}) \right\|_2^2 - L \sum_{k=2}^{\infty} \frac{(-1)^k}{k} \left[ \frac{1}{2} \text{Var}[\xi^{-2}] \right]^k$$

where $\Psi = \left\| \mathbf{\nabla}_{\mathbf{\mu}_c} \log p(\mathbf{W}|E[\Xi^{-2}]) \right\|_2^2 - \frac{1}{2E[\xi^2]}$.

The derivation is provided in the supplementary materials. We see that the gap depends on the variance of $1/\xi^2$ and the norm of the prior’s gradient w.r.t. the precision. While the latter term is likely manageable, the former term is not. In Figure 1 we plot $E[\xi^2]$ vs $\text{Var}[\xi^{-2}]$ (dashed lines) for the same scale distributions, and we see that either they asymptote as $E[\xi^2] \rightarrow 0$—which occurs for Gaussian noise (green) at $\sim 4$ and for inverse Nakagami noise (blue) at $\sim 0.5$—or are infinite for the whole range—Bernoulli (black) and Rayleigh (red). Thus, the series will not converge when the shrinkage is strong, only when $E[\xi^2]$ is well above one.

In summary, Proposition 2.1 shows that injecting noise into the likelihood relaxes the MAP estimate as a function of $\text{Var}[\xi]$ whereas Proposition 2.2 reveals noise in the Gaussian prior has a $\text{Var}[\xi^{-2}]$-order gap. When $E[\xi^2]$ is near zero, $\text{Var}[\xi^{-2}]$ explodes for all noise distributions considered (Figure 1), resulting in an impractical objective.

**Proposition 2.1: Expanded Parametrization Gap**

$$J_{\text{HP-GAP}} = \log p(\mathbf{y}, \{\mathbf{W}_i\}_{i=1}^{L+1}|\mathbf{X}) - E_{p(\xi)} \left[ \log p(\mathbf{y}, \{\mathbf{W}_i\}_{i=1}^{L+1}|\mathbf{X}, \{\Xi_i\}_{i=1}^{L}) \right]$$

$$= \log p(\mathbf{y}|\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^{L+1}) + \log p(\{\mathbf{W}_i\}_{i=1}^{L+1}) - E_{p(\xi)} \left[ \log p(\mathbf{y}|\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^{L+1}, \{\Xi_i\}_{i=1}^{L}) \right] - \log p(\{\mathbf{W}_i\}_{i=1}^{L+1})$$

$$= \log E_{p(\xi)} \left[ p(\mathbf{y}|\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^{L+1}, \{\Xi_i\}_{i=1}^{L}) \right] - E_{p(\xi)} \left[ \log p(\mathbf{y}|\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^{L+1}, \{\Xi_i\}_{i=1}^{L}) \right].$$

Expanding $E_{p(\xi)} \left[ p(\mathbf{y}|\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^{L+1}, \{\Xi_i\}_{i=1}^{L}) \right]$ around $E[\Xi]$, we have

$$E_{p(\xi)} \left[ p(\mathbf{y}|\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^{L+1}, \{\Xi_i\}_{i=1}^{L}) \right] \approx p(\mathbf{y}|\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^{L+1}, \{E[\Xi_i]\}_{i=1}^{L}) + \frac{1}{2} \text{Var}[\xi] Tr \left\{ \mathbf{\nabla}_{\mathbf{\mu}_c} p(\mathbf{y}|\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^{L+1}, \{E[\Xi_i]\}_{i=1}^{L}) \right\},$$

Figure 1. Noise Variances. The plot shows $\text{Var}[\xi]$ in solid lines and $\text{Var}[\xi^{-2}]$ in dashed lines for the noise distributions in Table 1 of the main text.
and dividing through by \( p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \) inside the log yields:

\[
\log \mathbb{E}_{p(\xi)} \left[ p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right] \approx \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) + \log \left\{ 1 + \frac{1}{2} \text{Var}[\xi] \text{Tr} \left\{ \nabla_{\mu_{\xi}}^2 \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right\} \right\}.
\]

Noticing that

\[
\nabla_{\mu_{\xi}}^2 p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) = p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \left( \nabla_{\mu_{\xi}} \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right)^2 + \text{Var}[\xi] \nabla_{\mu_{\xi}} \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L})
\]

allows the previous expression to simplify to

\[
\log \mathbb{E}_{p(\xi)} \left[ p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right] \approx \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) + \log \left\{ 1 + \frac{1}{2} \text{Var}[\xi] \text{Tr} \left\{ \nabla_{\mu_{\xi}}^2 \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right\} \right\}.
\]

where \( \Omega = \left| \left\{ \nabla_{\mu_{\xi}} \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right\} \right|^2 + \text{Tr} \left\{ \nabla_{\mu_{\xi}}^2 \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right\} \).

Next expanding the second term similarly around \( \mathbb{E}[\Xi] \), we have

\[
\mathbb{E}_{p(\xi)} \left[ \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right] \approx \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) + \frac{1}{2} \text{Var}[\xi] \text{Tr} \left\{ \nabla_{\mu_{\xi}}^2 \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right\}.
\]

Now we put both terms together while expanding the log \( \cdot \) around 1:

\[
\mathcal{J}_{\text{HP-GAP}} \approx \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) + \log 1 - \sum_{k=1}^{\infty} \frac{(-1)^k}{k} \left[ \frac{1}{2} \text{Var}[\xi] \Omega \right]^k
\]

\[
- \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) - \frac{1}{2} \text{Var}[\xi] \text{Tr} \left\{ \nabla_{\mu_{\xi}}^2 \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right\}
\]

\[
= -\sum_{k=1}^{\infty} \frac{(-1)^k}{k} \left[ \frac{1}{2} \text{Var}[\xi] \Omega \right]^k - \frac{1}{2} \text{Var}[\xi] \text{Tr} \left\{ \nabla_{\mu_{\xi}}^2 \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right\}
\]

\[
= \frac{1}{2} \text{Var}[\xi] \Omega - \frac{1}{2} \text{Var}[\xi] \text{Tr} \left\{ \nabla_{\mu_{\xi}}^2 \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right\} - \sum_{k=2}^{\infty} \frac{(-1)^k}{k} \left[ \frac{1}{2} \text{Var}[\xi] \Omega \right]^k
\]

\[
= \frac{1}{2} \text{Var}[\xi] \Omega - \frac{1}{2} \text{Var}[\xi] \text{Tr} \left\{ \nabla_{\mu_{\xi}}^2 \log p(y|X, \{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right\} - \frac{1}{8} \text{Var}[\xi] \Omega^2 + O(\text{Var}[\xi]).
\]

We assume that \( \text{Var}[\xi] \rightarrow 0 \) (i.e. strong shrinkage) and this gives an \( O(\text{Var}[\xi]) \) convergence.

**Proposition 2.2:** Hierarchical Parametrization Gap

\[
\mathcal{J}_{\text{HP-GAP}} = \log p(y, \{ W_i \}_{l=1}^{L+1}|X) - \mathbb{E}_{p(\xi)} \left[ \log p(y, \{ W_i \}_{l=1}^{L+1}|X, \{ \Xi \}_{l=1}^{L}) \right]
\]

\[
= \log p(y|X, \{ W_i \}_{l=1}^{L+1}) + \log \mathbb{E}_{p(\xi)} \left[ p(W|\{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right] - \log p(y|X, \{ W_i \}_{l=1}^{L+1}) - \mathbb{E}_{p(\xi)} \left[ \log p(W|\{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right]
\]

\[
= \log \mathbb{E}_{p(\xi)} \left[ p(W|\{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right] - \mathbb{E}_{p(\xi)} \left[ \log p(W|\{ W_i \}_{l=1}^{L+1}, \{ \Xi \}_{l=1}^{L}) \right]
\]

Next we reparametrize so that \( \Xi^{-2} \) denotes the Gaussian’s precision. We then expand the first term around \( \mathbb{E}[\Xi^{-2}] \):

\[
\mathbb{E}_{p(\xi)} \left[ p(W|\Xi^{-2}) \right] \approx p(W|\mathbb{E}[\Xi^{-2}]) + \frac{1}{2} \text{Var}[\xi^{-2}] \text{Tr} \left\{ \nabla_{\mu_{\xi^{-2}}}^2 p(W|\mathbb{E}[\Xi^{-2}]) \right\}.
\]
Dividing inside the log then allows us to write:

\[ L \log \mathbb{E}_{p(\xi)} \left[ p(W | \Xi^{-2}) \right] \approx L \log p(W | E | \Xi^{-2}) + L \log \left\{ 1 + \frac{1}{2} Var[\xi^{-2}] Tr \left\{ \nabla^2_{\mu_\xi} p(W | E | \Xi^{-2}) \right\} \right\}. \]

Noticing that

\[ \nabla^2_{\mu_\xi} p(W | E | \Xi^{-2}) = p(W | E | \Xi^{-2}) \left[ \left( \nabla_{\mu_\xi} \log p(W | E | \Xi^{-2}) \right)^2 + \nabla^2_{\mu_\xi} \log p(W | E | \Xi^{-2}) \right] \]

allows the previous expression to simplify to

\[ L \log \mathbb{E}_{p(\xi)} \left[ p(W | \Xi^{-2}) \right] \approx L \log p(W | E | \Xi^{-2}) + L \log \left\{ 1 + \frac{1}{2} Var[\xi^{-2}] \Psi \right\}, \]

where

\[ \Psi = ||\nabla_{\mu_\xi} \log p(W | E | \Xi^{-2})||^2 + Tr \left\{ \nabla^2_{\mu_\xi} \log p(W | E | \Xi^{-2}) \right\}. \]

Next we can write out the second term as

\[ L \mathbb{E}_{p(\xi)} \left[ \log p(W | E | \Xi^{-2}) \right] = \frac{-L}{2\sigma_0^2} \mathbb{E}[\xi^{-2}]||W||^2 + \frac{L}{2} \mathbb{E}[\log \xi^{-2}] - \frac{L}{2} \log 2\pi \]

\[ \approx \frac{-L}{2\sigma_0^2} \mathbb{E}[\xi^{-2}]||W||^2 + \frac{L}{2} \log \mathbb{E}[\xi^{-2}] - \frac{L}{2} \log 2\pi + \frac{L}{4} Var[\xi^{-2}] \frac{-1}{4 \mathbb{E}[\xi^{-2}]^2} \]

\[ = L \log p(W | E | \Xi^{-2}) - Var[\xi^{-2}] \frac{L}{4 \mathbb{E}[\xi^{-2}]^2}. \]

Rejoining the expressions, again expanding the log around 1, we have:

\[ J_{HP-GAP} \approx L \log p(W | E | \Xi^{-2}) + L \log 1 - L \sum_{k=1}^{\infty} \frac{(-1)^k}{k} \left[ \frac{1}{2} Var[\xi^{-2}] \Psi \right]^k - L \log p(W | E | \Xi^{-2}) + Var[\xi^{-2}] \frac{L}{4 \mathbb{E}[\xi^{-2}]^2} \]

\[ = -L \sum_{k=1}^{\infty} \frac{(-1)^k}{k} \left[ \frac{1}{2} Var[\xi^{-2}] \Psi \right]^k + Var[\xi^{-2}] \frac{L}{4 \mathbb{E}[\xi^{-2}]^2} \]

\[ = \frac{L}{2} Var[\xi^{-2}] \Psi + \frac{L}{2} Var[\xi^{-2}] \frac{1}{2 \mathbb{E}[\xi^{-2}]^2} - L \sum_{k=2}^{\infty} \frac{(-1)^k}{k} \left[ \frac{1}{2} Var[\xi^{-2}] \Psi \right]^k \]

\[ = \frac{L}{2} Var[\xi^{-2}] ||\nabla_{\mu_\xi} \log p(W | E | \Xi^{-2})||^2 + L \sum_{k=2}^{\infty} \frac{(-1)^k}{k} \left[ \frac{1}{2} Var[\xi^{-2}] \Psi \right]^k. \]

3. Variational EM Updates

3.1. ADD EM Updates

Below we derive the EM updates for the various hyper-priors considered. For all expressions, we assume the variational distribution on the weights factorizes, i.e. \( N(W; \phi) = \prod_i \prod_j N(w_{i,j}; \phi_{i,j}) \) where \( i \) denotes row and \( j \) column indices.

3.1.1. INVERSE GAMMA

We begin with the ELBO terms that depend on the variational variance \( \tau \):

\[ J_{ELBO}(\phi, \tilde{\tau}) = -KL [N(W; \phi) || N(W; \tilde{\tau})] + \log p(\tilde{\tau}) \]

\[ = -\frac{1}{2} \sum_i \sum_j \left[ \log \frac{\tilde{\tau}_i}{\sigma^2_{i,j}} + \frac{\sigma^2_{i,j}}{\tilde{\tau}_i} + \frac{\mu^2_{i,j}}{\tilde{\tau}_i} - 1 \right] + \log \Gamma^{-1}(\tilde{\tau}_i; \alpha, \beta) \]

\[ = -\frac{1}{2} \left[ \sum_i \sum_j \log \frac{\tilde{\tau}_i}{\sigma^2_{i,j}} + \sum_i \sum_j \left( \frac{\sigma^2_{i,j} + \mu^2_{i,j}}{\tilde{\tau}_i} - D \right) - \frac{\beta}{\tilde{\tau}_i} - (\alpha + 1) \log \tilde{\tau}_i + \log \frac{\beta^\alpha}{\Gamma(\alpha)} \right]. \]
Differentiating w.r.t. $\bar{\tau}_l$ and setting to zero, we have

\[
0 = \frac{\partial}{\partial \bar{\tau}_l} J_{\text{ELBO}}(\phi, \bar{\tau}_l)
= -\frac{1}{2} \left[ \sum_i \sum_j \frac{1}{\bar{\tau}_l} - \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} \right] + \frac{\beta}{\bar{\tau}_l^2} - \frac{\alpha + 1}{\bar{\tau}_l}
= \frac{D}{\bar{\tau}_l} - \sum_i \sum_j \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} - \frac{2\beta}{\bar{\tau}_l^2} + \frac{2\alpha + 2}{\bar{\tau}_l}
\]

\[
\frac{2\beta + \sum_i \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} = \frac{D + 2\alpha + 2}{\bar{\tau}_l}
\]

\[
\bar{\tau}_l^* = \frac{2\beta + \sum_i \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\frac{D + 2\alpha + 2}{\bar{\tau}_l}} = \frac{D + 2\alpha + 2}{\bar{\tau}_l} + \frac{2}{\eta^2 + \bar{\tau}_l}
\]

where $D$ are the number of dimensions (i.e. parameters) in $W$.

### 3.1.2. Half-Cauchy

Again starting with the ELBO, we have

\[
0 = \frac{\partial}{\partial \bar{\tau}_l} J_{\text{ELBO}}(\phi, \bar{\tau}_l)
= -\frac{1}{2} \frac{\partial}{\partial \bar{\tau}_l} \sum_i \sum_j \left[ \log \frac{\bar{\tau}_l}{\sigma_{i,j}} + \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} - 1 \right] + \frac{\partial}{\partial \bar{\tau}_l} \log C^+(\sqrt{\bar{\tau}_l}; 0, \eta)
= -\frac{1}{2} \left[ \sum_i \sum_j \frac{1}{\bar{\tau}_l} - \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} \right] + \frac{\partial}{\partial \bar{\tau}_l} \log \frac{2}{\pi \eta (1 + \bar{\tau}_l / \eta^2)}
= \frac{D}{\bar{\tau}_l} - \sum_i \sum_j \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} + \frac{2}{\eta^2 + \bar{\tau}_l}
= \frac{2\bar{\tau}_l}{\eta^2 + \bar{\tau}_l} - \sum_i \sum_j \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l} + D.
\]

Solving\(^{1}\) the above equation for $\bar{\tau}_l$ gives for a positive solution:

\[
\bar{\tau}_l^* = \frac{M - \eta^2 D + \sqrt{M^2 + (2D + 8)\eta^2 M + \eta^4 D^2}}{2D + 4} \quad \text{where} \quad M = \sum_i \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2.
\]

$D$ is again the dimensionality, and $\eta$ is the half-Cauchy’s scale.

\(^{1}\)We plugged the equation into Wolfram Alpha.
3.1.3. LOG-UNIFORM

Again beginning with ELBO, we have

\[
0 = \frac{\partial}{\partial \tau_i} \mathcal{J}_{\text{ELBO}}(\phi, \bar{\tau}_i)
= -\frac{1}{2} \left[ \sum_i \sum_j \frac{1}{\tau_i} - \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\tau_i} \right] + \frac{\partial}{\partial \tau_i} \log \frac{c}{\tau_i}
= \frac{D}{\bar{\tau}_i} - \frac{\sum_i \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_i} + \frac{2}{\bar{\tau}_i}
\]

\[
\bar{\tau}_i^* = \frac{\sum_i \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{D + 2}
\]

Notice that this update is the same as the inverse Gamma’s as \(\alpha \to 0\) and \(\beta \to 0\).

3.2. ARD-ADD EM Updates

For approximate posterior inference for the joint ARD-ADD prior, we assume the posterior approximation

\[
p(W, \lambda, \tau | y, X) \approx q(W; \phi) q(\lambda) q(\tau) = N(W; \mu_\phi, \text{diag}\{\Sigma_\phi\}) \delta[\bar{\lambda}_i] \delta[\bar{\tau}_i]
\]

where \(\phi = \{\mu_\phi, \text{diag}\{\Sigma_\phi\}\}, \bar{\lambda}_i,\) and \(\bar{\tau}_i\) are the variational parameters. The ELBO for this approximation is

\[
\log p(y|X) \geq \mathbb{E}_{q(W)} [\log p(y|X, W)] - \mathbb{E}_{q(\lambda)} \mathbb{E}_{q(\tau)} \text{KL} [q(W; \phi)||p(W|\lambda, \tau)] - \text{KL} [q(\lambda)||p(\lambda)] - \text{KL} [q(\tau)||p(\tau)]
= \mathbb{E}_{q(W)} [\log p(y|X, W)] - \text{KL} [N(W; \phi)||N(W|\bar{\lambda}_i, \bar{\tau}_i)] - \text{KL} [\delta[\bar{\lambda}_i]||p(\lambda)] - \text{KL} [\delta[\bar{\tau}_i]||p(\tau)]
= \mathbb{E}_{q(W)} [\log p(y|X, W)] - \text{KL} [N(W; \phi)||N(W|\bar{\lambda}_i, \bar{\tau}_i)] + \log p(\bar{\lambda}_i) + \log p(\bar{\tau}_i) + C
\]

where \(C = \mathbb{H}[\delta[\bar{\lambda}_i]] + \mathbb{H}[\delta[\bar{\tau}_i]]\) is again a constant. Next we must find solutions for both \(\bar{\lambda}_i\) and \(\bar{\tau}_i\). Again we can differentiate the ELBO and set to zero:

\[
\frac{\partial}{\partial \lambda_i} \mathcal{J}_{\text{ELBO}}(\phi, \bar{\lambda}_i, \bar{\tau}_i) = -\frac{\partial}{\partial \lambda_i} \text{KL} [N(W; \phi)||N(W|\bar{\lambda}_i, \bar{\tau}_i)] + \frac{\partial}{\partial \lambda_i} \log p(\bar{\lambda}_i) = 0,
\]

\[
\frac{\partial}{\partial \tau_i} \mathcal{J}_{\text{ELBO}}(\phi, \bar{\lambda}_i, \bar{\tau}_i) = -\frac{\partial}{\partial \tau_i} \text{KL} [N(W; \phi)||N(W|\bar{\lambda}_i, \bar{\tau}_i)] + \frac{\partial}{\partial \tau_i} \log p(\bar{\tau}_i) = 0.
\]

We denote the solutions as \(\bar{\lambda}_i^*\) and \(\bar{\tau}_i^*\), deriving them for the three hyper-priors below. Again, we update \(q(W; \phi)\) via gradient ascent:

\[
\frac{\partial}{\partial \phi} \mathcal{J}_{\text{ELBO}}(\phi, \bar{\lambda}_i^*, \bar{\tau}_i^*) = \frac{\partial}{\partial \phi} \mathbb{E}_{q(W; \phi)} [\log p(y|X, W)] - \frac{\partial}{\partial \phi} \text{KL} [N(W; \phi)||N(W|\bar{\lambda}_i^*, \bar{\tau}_i^*)].
\]
where $D_i$ is the number of parameters in the $i$th row (i.e. $\lambda$’s corresponding row) of the weight matrix $W$. Furthermore, notice that the sum over posterior parameters is across only columns ($j$ index).

Moving on the ADD parameter, we begin with the second line of Equation 3:

$$0 = \frac{\partial}{\partial \bar{\tau}_i} J_{\text{ELBO}}(\phi, \bar{\lambda}_i, \bar{\tau}_i)$$

$$= -\frac{1}{2} \left[ \sum_i \sum_j \frac{1}{\lambda_i} \left( \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\lambda_i^2 \bar{\tau}_i} \right) - \frac{2\beta}{\lambda_i^2} + 2\alpha + 2 \frac{\bar{\tau}_i}{\lambda_i} \right] + \frac{\partial}{\partial \bar{\tau}_i} \log \Gamma^{-1}(\bar{\tau}_i; \alpha, \beta)$$

$$\bar{\tau}_i^* = \frac{2\beta \bar{\tau}_i + 2\sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{D + 2\alpha + 2} \bar{\tau}_i$$

3.2.2. HALF-CAUCHY

Starting with the first line of Equation 3, we have

$$0 = \frac{\partial}{\partial \lambda_i} J_{\text{ELBO}}(\phi, \bar{\lambda}_i, \bar{\tau}_i)$$

$$= -\frac{1}{2} \left[ \sum_i \sum_j \frac{1}{\lambda_i} \left( \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\lambda_i^2 \bar{\tau}_i} \right) - \frac{2\beta}{\lambda_i^2} + 2\alpha + 2 \frac{\bar{\tau}_i}{\lambda_i} \right] + \frac{\partial}{\partial \lambda_i} \log C^+(\sqrt{\lambda_i}; 0, \eta)$$

$$\lambda_i^* = \frac{2\bar{\lambda}_i}{\eta^2 + \lambda_i} - \frac{2\lambda_i}{\eta^2 + \lambda_i} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2$$

$$= \frac{2\bar{\lambda}_i}{\eta^2 + \lambda_i} \frac{\bar{\tau}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{D_i} + D_i$$
where $D_i$ is the number of parameters in the $i$th row. This final expression is of the same form we solved for ADD and therefore has the same solution with appropriately adjusted constants:

$$
\lambda_i^* = \frac{M_i - \eta^2 D_i + \sqrt{M_i^2 + (2D_i + 8)\eta^2 M_i + \eta^4 D_i^2}}{2D_i + 4}
$$

where $M_i = \tilde{\tau}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2$.

Moving on the ADD parameter, we consider the second line of Equation 3:

$$
0 = \frac{\partial}{\partial \tilde{\tau}_i} J_{ELBO}(\phi, \lambda_i, \tilde{\tau}_i)
$$

$$
= -\frac{1}{2} \frac{\partial}{\partial \tilde{\tau}_i} \sum_i \sum_j \left[ \log \frac{\lambda_i \tilde{\tau}_j}{\sigma_{i,j}^2} + \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\lambda_i \tilde{\tau}_j} - 1 \right] + \frac{\partial}{\partial \tilde{\tau}_i} \log C^+(\sqrt{\tilde{\tau}_i}; 0, \eta)
$$

$$
= \sum_i \sum_j \frac{1}{\tilde{\tau}_i} \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\lambda_i \tilde{\tau}_j^2} + \frac{2}{\eta^2 + \tilde{\tau}_i}
$$

$$
= \frac{D}{\tilde{\tau}_i} - \sum_i \tilde{\lambda}_i^{-1} \sum_j \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\tilde{\tau}_j} - \frac{2}{\eta^2 + \tilde{\tau}_i}
$$

$$
= \frac{2\tilde{\tau}_i}{\eta^2 + \tilde{\tau}_i} - \sum_i \tilde{\lambda}_i^{-1} \sum_j \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\tilde{\tau}_j} + D.
$$

Again solving the same equation yields the solution:

$$
\tilde{\tau}_i^* = \frac{M_\lambda - \eta^2 D + \sqrt{M_\lambda^2 + (2D + 8)\eta^2 M_\lambda + \eta^4 D^2}}{2D + 4}
$$

where $M_\lambda = \sum_i \tilde{\lambda}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2$.

3.2.3. LOG-UNIFORM

As mentioned earlier, the solution for the log-uniform distribution can be attained in the limit of the inverse Gamma. Plugging $\alpha = 0$ and $\beta = 0$ into the inverse gamma’s solutions we obtain:

$$
\lambda_i^* = \frac{\sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\tilde{\tau}_i(D_i + 2)}
$$

and

$$
\tilde{\tau}_i^* = \frac{\sum_i \tilde{\lambda}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{D + 2}.
$$

4. Experimental Details

Below we provide more details on our experimental set-up. For all experiments, we used following settings, replicating the set-up of Gal & Ghahramani (2016). Optimization was done using Adam (Kingma & Ba, 2014) with a learning rate of .001 and other parameters kept at Tensorflow defaults. Training was done for 4500 epochs using batch sizes of 32 and 10 samples when calculating Monte Carlo expectations. The UCI regression data sets were divided into 20 randomized 90% - 10% train-test splits. Results are reported by training and evaluating on each split and reporting the average test metric. Test set evaluations were done by drawing 500 MC samples. The training split was standardized to zero mean and unit variance, and predictions were made by normalizing the test input based on these train set statistics and then applying the inverse transform to the prediction. RMSE and likelihood were calculated after applying the inverse transform.

Importance Sampling Experiments All results in Table 2 of the main text used a one-hidden-layer neural network with 50 hidden units and ReLU activations. We used the dropout rates and likelihood precisions reported in Table 1. These parameters were obtained from the validation set results reported by Gal & Ghahramani (2016) at: https://github.com/yaringal/DropoutUncertaintyExps/tree/master/UCI_Datasets.

Variational EM Experiments The ARD, ADD, and ARD-ADD results in Table 3 of the main text used neural networks with two hidden layers with one skip connection between them. ReLU activations were again used exclusively. ARD was applied to the first and last weight matrices—since ADD can’t be applied without a skip connection—and thus the
Table 1. The table above reports the dropout rate and likelihood precision used for each experiment. We use the same parameters as Gal & Ghahramani (2016).

5. Tensorflow Implementations

We have provided Python code to replicate the results presented in Tables 2 and 3 of the main text. We include the Yacht data set for the user’s convenience; the other data sets can be downloaded from Gal & Ghahramani (2016)’s GitHub repository or from the UCI repository (Dheeru & Karra Taniskidou, 2017). Below we give implementation snippets for the paper’s core algorithmic contributions.

5.1. Implementation of Tail-Adaptive Importance Weights

```python
log_likelihoods = # matrix of size Batch_Size x Number_of_MC_Samples

# Sort the samples
log_likelihoods_sorted, _ = tf.nn.top_k(log_likelihoods, k=n_samples, sorted=True)

# Use tf.range to provide ranks
gamma_vals = n_samples / (tf.expand_dims(tf.range(n_samples, dtype=tf.float32), 0) + 1.)

# Normalize and treat weight as a constant
normalized_weights = tf.stop_gradient(gamma_vals / tf.reduce_sum(gamma_vals, axis=1, keep_dims=True))

final_ell_term = tf.reduce_sum(normalized_weights * log_likelihoods_sorted, reduction_indices=1, keep_dims=True)

# Return final expected log-likelihood term s.t. gradient will be w * d_ell
return tf.reduce_mean(final_ell_term)
```

5.2. EM Scale Updates

```python
def ard_scale_update(prior_id, prior_params, mu, std):
    param_count = mu.get_shape().as_list()[1]
    moments = tf.reduce_sum(mu*mu + std*std, reduction_indices=1, keep_dims=True)
    if prior_id == "inv_gamma":
```

差異在性能上是由于其對隱層到隱層權重的影響。我們使用flipout (Wen et al., 2018) 來降低Monte Carlo期望（10個樣本）的方差。所有比例尺度的先驗都設置為相同的分布，我們在子集上選擇此分布進行初步實驗。Protein UCI數據集（Dheeru & Karra Taniskidou, 2017）。我們嘗試的先驗有：\(\Gamma^{-1}(0, 0)\) / log-uniform, \(\Gamma^{-1}(1, 1)\), \(\Gamma^{-1}(3, 3)\), \(\Gamma^{-1}(5, 5)\), \(\Gamma^{-1}(1, 3)\), \(\Gamma^{-1}(1, 5)\), \(\Gamma^{-1}(1, 10)\), \(\mathcal{C}^+(0, 5)\), \(\mathcal{C}^+(0, 1)\), \(\mathcal{C}^+(0, 2.5)\), \(\mathcal{C}^+(0, 5)\), \(\mathcal{C}^+(0, 10)\)。\(\Gamma^{-1}(3, 3)\)表現最好，我們使用它進行所有實驗。這表明這一個先驗在不同的數據集上具有很好的泛化能力，這證明了EM算法的健壯性。

5. Tensorflow Implementations

我們已經提供Python代碼來複製主要文本中表2和表3中呈現的結果。我們包括了Yacht數據集以供用戶使用，其他數據集可以从Gal & Ghahramani (2016)的GitHub存儲庫獲取或從UCI存儲庫（Dheeru & Karra Taniskidou, 2017）。下面我們提供實現代碼片段，以示該文章的核心算法貢獻。

5.1. Implementation of Tail-Adaptive Importance Weights

```python
log_likelihoods = # matrix of size Batch_Size x Number_of_MC_Samples

# Sort the samples
log_likelihoods_sorted, _ = tf.nn.top_k(log_likelihoods, k=n_samples, sorted=True)

# Use tf.range to provide ranks
gamma_vals = n_samples / (tf.expand_dims(tf.range(n_samples, dtype=tf.float32), 0) + 1.)

# Normalize and treat weight as a constant
normalized_weights = tf.stop_gradient(gamma_vals / tf.reduce_sum(gamma_vals, axis=1, keep_dims=True))

final_ell_term = tf.reduce_sum(normalized_weights * log_likelihoods_sorted, reduction_indices=1, keep_dims=True)

# Return final expected log-likelihood term s.t. gradient will be w * d_ell
return tf.reduce_mean(final_ell_term)
```

5.2. EM Scale Updates

```python
def ard_scale_update(prior_id, prior_params, mu, std):
    param_count = mu.get_shape().as_list()[1]
    moments = tf.reduce_sum(mu*mu + std*std, reduction_indices=1, keep_dims=True)
    if prior_id == "inv_gamma":
```
Supplementary Materials

```python
prior_opt_scale = tf.sqrt((2. * prior_params["param2"] + moments) / (param_count + 2. * prior_params["param1"] + 2.))

elif prior_id == "half_cauchy":
    prior_opt_scale = tf.sqrt((moments - prior_params["param1"]**2 * param_count + tf.sqrt(moments**2 + (2 * param_count + 8) * prior_params["param1"]**2 * moments + prior_params["param1"]**4 * param_count**2)) / (2 * param_count + 4))

# Log-uniform implemented by just passing alpha=beta=0 to inv Gamma
return prior_opt_scale * tf.ones_like(std)

def add_scale_update(prior_id, prior_params, mu, std):
    param_count_rows = mu.get_shape().as_list()[1]
    param_count_cols = mu.get_shape().as_list()[0]
    moments = tf.reduce_sum(mu * std + 2, reduction_indices=1, keep_dims=True)

    if prior_id == "inv_gamma":
        prior_opt_scale = tf.sqrt((2. * prior_params["param2"] + moments) / (param_count + 2. * prior_params["param1"] + 2.))
        opt_lambda = tf.sqrt((2. * prior_params["param1"] + old_tau + moments) / (old_tau * 2. + prior_params["param1"] + param_count_cols + old_tau * 2))
        opt_tau = tf.sqrt((2. * prior_params["param2"] + std + param_count(rows + 2 + 2 * prior_params["param1"]) / (param_count_cols + param_count_rows + 2 + 2 + prior_params["param1"]) )

    if prior_id == "half_cauchy":
        moments_lamb = tf.reduce_sum((1. / opt_lambda) * moments)
        param_count = param_count_rows * param_count_cols
        opt_lamb = tf.sqrt((moments_lamb - prior_params["param1"]**2 * param_count_cols + tf.sqrt(moments_lamb**2 + (2 * param_count + 8) * prior_params["param1"]**2 * moments_lamb + prior_params["param1"]**4 * param_count**2)) / (2 * param_count + 4))

    # Log-uniform implemented by just passing alpha=beta=0 to inv Gamma
    return opt_lambda * opt_tau * tf.ones_like(std)

 References


