# Appendices

# A Further Details from the Harmonic Analysis Framework for Boolean Functions

## A.1 Discrete derivative

For a Boolean function $f$ the discrete derivative on the $i$-th latent dimension with a basis function $\phi_i$ is defined as

$$D_{\phi_i} f(z) = \sigma_i \frac{f(z_1, ..., z_i = +1, ..., z_n) - f(z_1, ..., z_i = -1, ..., z_n)}{2}. \tag{1}$$

The Fourier expansion of the discrete derivative equals $D_{\phi_i} f(z) = \sum_{S \ni i} \hat{f}^{(p)}(S) \phi_{S \setminus i}(z)$, The $i$-th discrete derivative is independent of $z_i$.

# B Harmonic Analysis of Existing Gradient Estimates

## B.1 Proof of Lemma 1

*Proof.* We first derive a relation between the true gradient $\partial_{p_i} \mathbb{E}_{p(z)}[f(z)]$ and the degree-1 Fourier coefficients $\hat{f}^{(p)}(i)$. This is an extension of the Margulis-Russo formula (Margulis, 1974; Russo, 1982; O'Donnell, 2014). We show that

$$\partial_{p_i} \mathbb{E}_{p(z)}[f(z)] = \frac{2}{\sigma_i} \hat{f}^{(p)}(i).$$

We follow O'Donnell (2014, §8.4). We work with two representations of the Boolean function $f$. The first is the Fourier expansion of $f$ under the uniform Bernoulli distribution. This is also the representation obtained by expressing $f$ as a polynomial in $z_i$. Since the domain of the function $f$ is the Boolean cube, the polynomial representation is multilinear. That is $f(z) = \sum_{S \subseteq [n]} \hat{f}(S) \prod_{j \in S} z_j$. To avoid confusion and to differentiate the representation from the Boolean function, we use $f^{(u)}(z)$ to denote this representation in the following. Note that since this representation is a polynomial it is defined over any input in $\mathbb{R}^n$. In particular,

$$\mathbb{E}[f(z)] = \mathbb{E}[\sum_{S \subseteq [n]} \hat{f}(S) \prod_{j \in S} z_j] = \sum_{S \subseteq [n]} \hat{f}(S) \prod_{j \in S} \mathbb{E}[z_j] = f^{(u)}(\mu_1, \ldots, \mu_n)$$

The second representation we use is the Fourier expansion of the Boolean function $f$ under $p(x)$. We denote this by $f^{(p)}$.

The following relation follows from the fact that when working with the Fourier representation, $f(z)$ is multilinear, $\mathbb{E}_{p(z)}[z_i] = \mu_i$ and the linearity of expectation.

$$\mathbb{E}_{p(z)}[f^{(p)}(z_1, \ldots, z_n)] = \mathbb{E}_{p(z)}[f(z_1, \ldots, z_n)] = f^{(u)}(\mu_1, \ldots, \mu_n). \tag{2}$$

As the partial derivative of $f^{(u)}$ w.r.t. $\mu_i$ is equivalent to discrete derivative of $f^{(u)}$ w.r.t. $z_i$, $\partial_{\mu_i} f^{(u)}(\mu) = D_{z_i} f^{(u)}(\mu)$, and keeping in mind that $\phi_i = (z_i - \mu_i)/\sigma_i$, we have that

$$\begin{aligned} D_{z_i} f^{(u)}(\mu) &= \mathbb{E}_{p(z)}[D_{z_i} f^{(p)}(z_1, \ldots, z_n)] && \text{(from (2))} && (3) \\ &= \frac{1}{\sigma_i} \mathbb{E}_{p(z)}[D_{\phi_i} f^{(p)}(z_1, \ldots, z_n)] && \text{(set } \phi_i = (z_i - \mu_i)/\sigma_i \text{, then chain rule)} && (4) \\ &= \frac{1}{\sigma_i} \hat{f}^{(p)}(i) && && (5) \end{aligned}$$

We then note that the discrete derivative of $f$ w.r.t. $z_i$, $D_{z_i} f^{(u)}(\mu)$, from the left hand side of (3), is equivalent to the partial derivative of $f$ w.r.t. $\mu_i$, $\partial_{\mu_i} f^{(u)}(\mu)$.

$$\begin{aligned}
\frac{1}{\sigma_i}\hat{f}^{(p)}(i) &= D_{z_i} f^{(u)}(\mu) = \partial_{\mu_i} f^{(u)}(\mu) && (6)\\
&= \frac{1}{2}\partial_{p_i} f^{(u)}(\mu) && \text{(set } \mu_i = 2p_i - 1\text{, then chain rule)} \quad (7)\\
&= \frac{1}{2}\partial_{p_i} \mathbb{E}_{p(z)}[f^{(p)}(z)] && \text{(from (2))} \quad (8)
\end{aligned}$$

We then note that the right hand side in (8) is $\frac{1}{2}$ times the true gradient.

**Rest of the proof of Lemma 1.** We derive the Taylor expansions for the true gradient as well as the Straight-Through gradient estimator. Then, we prove the lemma by comparing the two Taylor expansions.

By expanding the function $f$ in terms of its $\phi_S$ basis and focusing on the $i$-th dimension, we have that

$$\hat{f}^{(p)}(i)\phi_i = \hat{f}^{(p)}(i)\frac{z_i - \mu_i}{\sigma_i} = \hat{f}^{(p)}(i)\frac{z_i}{\sigma_i} - \hat{f}^{(p)}(i)\frac{\mu_i}{\sigma_i} \tag{9}$$

The first term, $\hat{f}^{(p)}(i)\frac{z_i}{\sigma_i}$, is the term corresponding to $\{i\}$ in the Fourier expansion of $f$ under $p^{i\to 1/2}(z)$. That is

$$\hat{f}^{(p\to 1/2)}(i) = \frac{1}{\sigma_i}\hat{f}^{(p)}(i) \tag{10}$$

This follows from the fact that when moving from $p(z)$ to $p^{i\to 1/2}(z)$, *(i)* we have that $\phi_i = z_i$, and *(ii)* no other term under the $p(z)$ expansion contributes to the $z_i$ term under the $p^{i\to 1/2}(z)$ expansion.

The true gradient for the $i$-th dimension is given by

$$\partial_{p_i} \mathbb{E}_{p(z)}[f(z)] = 2\frac{\hat{f}^{(p)}(i)}{\sigma_i} = 2\hat{f}^{(p\to 1/2)}(i) \tag{11}$$

Next, we will derive the Taylor expansions of the true and the Straight-Through gradients. The Taylor expansion of $f(z)$ for $z_i$ around 0 is

$$f(z) = c_0 + c_1 z_i + c_2 z_i^2 + c_3 z_i^3 + c_4 z_i^4 + c_5 z_i^5 + ..., \tag{12}$$

where $c_k k! = \partial^k_{z_i} f(z)|_{z_i=0}$ are the Taylor coefficients. All $c_k$ are a function of $z_j, j \neq i$.

Let's first focus on the true gradient. Since we work with binary ±1 values, we have that $z_i^k = 1$ for even $k$ and $z_i^k = z_i$ for odd $k$. This will influence the even and the odd terms of the Taylor expansions. Specifically, for the Taylor expansion of the true gradient we have from (11) that

$$\begin{aligned}
\partial_{p_i}[\mathbb{E}_{p(z)}[f(z)]] = 2\hat{f}^{(p\to 1/2)}(i) &= 2\,\mathbb{E}_{p^{i\to 1/2}(z)}[(c_0 + c_1 z_i + c_2 z_i^2 + \cdots)z_i] && (13)\\
&= 2\,\mathbb{E}_{p^{i\to 1/2}(z)}[c_0 z_i + c_1 z_i^2 + c_i z_i^3 + \cdots] && (14)\\
&= 2\,\mathbb{E}_{p^{i\to 1/2}(z)}[c_0 z_i + c_1 + c_2 z_i + c_3 + \cdots] && (z_i^{2j} = 1, z_i^{2j+1} = z_i, \mathbb{E}_{p^{i\to 1/2}(z)}[z_i] = 0) \quad (15)\\
&= 2\,\mathbb{E}_{p(z_{\setminus i})}[c_1 + c_3 + c_5 + \cdots] && (16)
\end{aligned}$$

The expression in (16) implies that the true gradient with respect to the $p_i$ is the expected sum of the odd Taylor coefficients. Here we note that the although final expression in (16) can also be derived by a finite difference method, it does not make explicit, as in (15), the dependence on $z_i$ and $\mu_i$ of the term inside the expectation.

Now, let's focus on the Straight-Through gradient. Taking the derivative of the Taylor expansion in (12) w.r.t. to $z_i$, we have

$$\partial_{z_i} f(z) = c_1 + 2c_2 z_i + 3c_3 z_i^2 + 4c_4 z_i^3 + 5c_5 z_i^4 + ... \tag{17}$$

The Straight-Through gradient is the expectation of (17) in the $i$-th dimension, that is

$$\mathbb{E}_{p(z)}[\partial_{z_i} f(z)] = \mathbb{E}_{p(z)}[c_1 + 2c_2 z_i + 3c_3 z_i^2 + 4c_4 z_i^3 + 5c_5 z_i^4 + \cdots] \tag{18}$$

$$= \mathbb{E}_{p(z_{\setminus i})}[c_1 + 3c_3 + 5c_5 + \cdots] + \mathbb{E}_{p(z_{\setminus i})}[2c_2 + 4c_4 + \cdots]\mu_i \quad (z_i^{2j} = 1, z_i^{2j+1} = z_i, \mathbb{E}_{p(z_i)}[z_i] = \mu_i) \tag{19}$$

By comparing the expansion of the Straight-Through gradient in (19) and the expansion of the true gradient in (16),

$$bias^{(p)}(g^i_{\mathrm{ST}}) = \mathbb{E}_{p(z)}[\partial_{z_i} f(z)] - \partial_{p_i} \mathbb{E}_{p(z)}[f(z)] \tag{20}$$

$$= \mathbb{E}_{p(z_{\setminus i})}\left[\sum_{k=2j+1, j>0} (k-2)c_k\right] + \mathbb{E}_{p(z_{\setminus i})}\left[\sum_{k=2j, j>0} kc_k\right]\mu_i. \tag{21}$$

Taking the expectation in (18) under $p^{i\to 1/2}$ causes the final term in (21) to vanish leaving

$$bias^{(p^{i\to 1/2})}(g^i_{\mathrm{ST}}) = \mathbb{E}_{p(z_{\setminus i})}\left[\sum_{k=2j+1, j>0} (k-2)c_k\right]. \tag{22}$$

□

# C Low-bias Gradient Estimates

## C.1 Lowering Bias by Representation Scaling

The Fourier basis does not depend on the particular input representation and any two-valued set, say $\{-t, t\}$ can be used as the Boolean representation. The choice of a representation, however, does affect the bias as we show next. As a concrete example, we let our input representation be $z_i \in \{-1/2, 1/2\}^n$, where $p_i = p(z_i = +1/2)$. While we can change the input representation like that, in general the Fourier coefficients will be different than for the $\{-1, +1\}$ representation. Letting $h(z_i) = 2z_i \in \{-1, 1\}$, the functions $\phi_i$ are now given as $\phi_i = \frac{h(z_i) - \mu_i}{\sigma_i}$.

Next, we write the Taylor series of $f$ in terms of $h(z_i)$,

$$f(z) = c_0 + c_1 z_i + c_2 z_i^2 + c_3 z_i^3 + c_4 z_i^4 + c_5 z_i^5 + \cdots \tag{23}$$

$$= c_0 + \frac{c_1}{2}h(z_i) + \frac{c_2}{2^2}h(z_i)^2 + \frac{c_3}{2^3}h(z_i)^3 + \frac{c_4}{2^4}h(z_i)^4 + \frac{c_5}{2^5}h(z_i)^5 + \cdots \tag{24}$$

Under the $p^{i\to 1/2}$ distribution, we still have that $\mathbb{E}_{p\to 1/2}[h(z_i)] = 0$ and the degree-1 Fourier coefficients are:

$$\hat{f}^{(p\to 1/2)}(i) = \mathbb{E}_{p^{i\to 1/2}(z)}[f(z)h(z_i)] = \mathbb{E}_{p^{i\to 1/2}(z)}\left[\frac{c_1}{2} + \frac{c_3}{2^3} + \frac{c_5}{2^5} + \cdots\right] \tag{25}$$

In (25) we still get the odd terms $c_1, c_3$ albeit decayed by inverse powers of 2. Following the same process as for the unscaled straight-through gradient, we have that

$$\frac{1}{2}\mathbb{E}_{p^{i\to 1/2}}[\partial_{z_i} f(z)] = \mathbb{E}_{p^{i\to 1/2}}\left[\frac{c_1}{2} + \frac{3c_3}{2^3} + \frac{5c_5}{2^5} + \cdots\right] \tag{26}$$

# D Efficiency Comparison

Table 1: Wall clock times for various gradient estimators on MNIST.

| **Method** | **#Eval.** | **Walltime in sec./Epoch** | |
|---|---|---|---|
| | | 2 layers | 5 layers |
| REBAR | 3 | 45.3 | 205.15 |
| ST | 1 | 2.86 | 4.96 |
| Gumbel | 1 | 3.27 | 6.14 |
| DARN | 1 | 2.96 | 5.36 |
| FouST | 1 | 3.1 | 5.67 |

# E Extra Experiments

## E.1 Experiments on MNIST

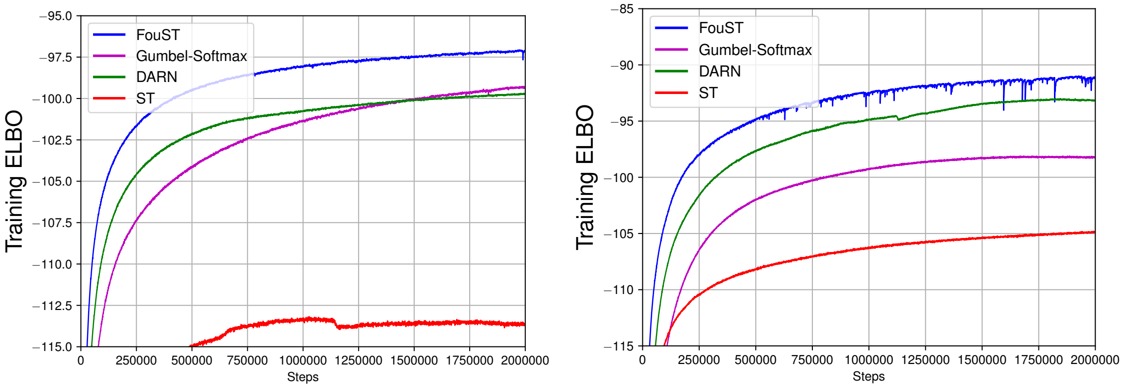

Figure 1: Training ELBO for the one (left) and two (right) stochastic layer nonlinear models on MNIST

Table 2: Test set performance with increasing stochastic depth on MNIST.

| Method | Stochastic Layers | Hidden Units per Layer | Test ELBO |
|---|---|---|---|
| Rebar | 2 | 200 | -94.43 |
| FouST | 2 | 200 | -91.95 |
| FouST | 3 | 200 | -89.11 |
| FouST | 8 | 500 | -87.31 |
| FouST | 20 | 500 | -87.86 |

## E.2 Ablation Experiments

To further judge the effect of our proposed modifications to Straight-Through, we performed ablation experiments where we separately applied scaling and noise to the importance-corrected Straight-Through. These experiments were performed on the single stochastic layer MNIST and OMNIGLOT models.

The results of the ablation experiments are shown in figure 2. From the figure it can be seen that scaling alone improves optimization in both cases and noise alone helps in the case of MNIST. Noise alone results in a worse ELBO in the case of OMNIGLOT, but gives an improvement when combined with scaling. From these results we conclude that the proposed modifications are effective.

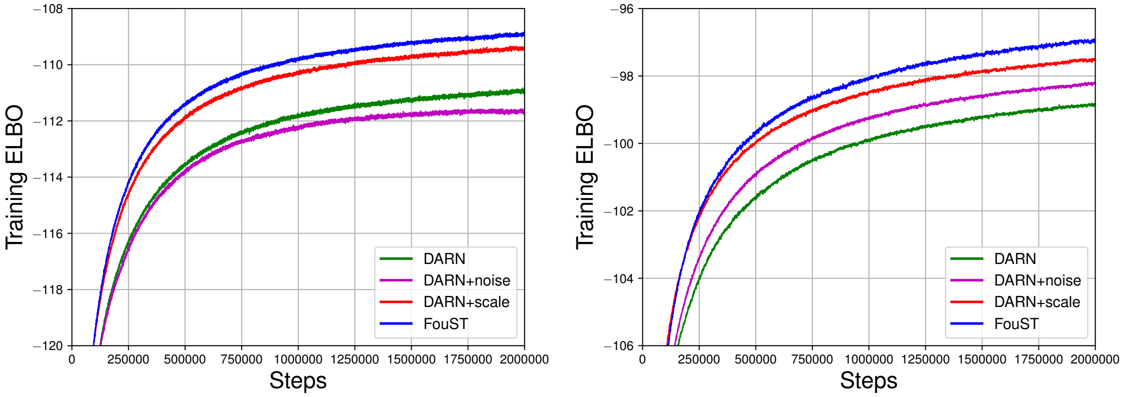

Figure 2: Ablations for the one stochastic layer nonlinear model on OMNIGLOT (left) and MNIST (right)

## E.3 Experiments on mini-ImageNet

We evaluate FouST on more complex neural networks with deeper and wider stochastic layers. We perform experiments with convolutional architectures on the larger scale and more realistic mini-ImageNet (Vinyals et al., 2016). As the scope of this work is not architecture search, we present two architectures inspired from residual networks (He et al., 2016) of varying stochastic depth and width. The first one is a wide S-ResNet, S-ResNet-40-2-800, and has 40 deterministic (with encoder and decoder combined), 2 stochastic layers, and 800 channels for the last stochastic layer. The second, S-ResNet-80-11-256, is very deep with 80 deterministic and 11 stochastic layers, and a last stochastic layer with 256 channels. Architecture details are given in Appendix F.2. In this setup, training with existing unbiased estimators is intractable.

We present results in Fig. 3. We compare against DARN, since we were unable to train the models with Gumbel-Softmax.

We observe that FouST is able to achieve better training ELBO's in both cases. We conclude that FouST allows for scaling up the complexity of stochastic neural networks in terms of stochastic depth and width.

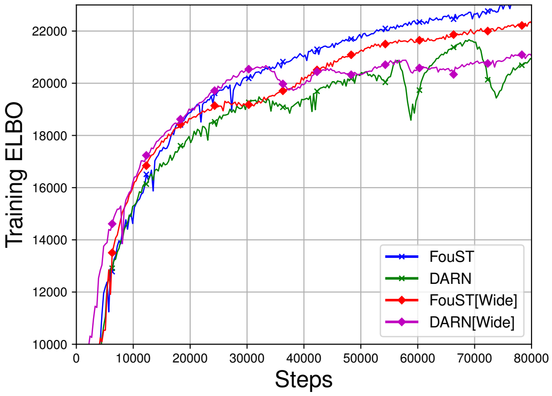

Figure 3: Training ELBO on mini-ImageNet

# F Architectures Used in the experiments

## F.1 Architectures for MNIST and Omniglot

The encoder and decoder networks in this case are MLP's with one or more stochastic layers. Each stochastic layer is preceded by 2 deterministic layers with a tanh activation function.

We chose learning rates from $\{1 \times 10^{-4}, 2 \times 10^{-4}, 4 \times 10^{-4}, 6 \times 10^{-4}\}$, Gumbel-Softmax temperatures from $\{0.1, 0.5\}$, and noise interval length for FouST from $\{0.1, 0.2\}$.

## F.2 Architectures for CIFAR-10 and mini-ImageNet

For these dataset we use a stochastic variant or ResNets (He et al., 2016). Each network is composed of *stacks* of *layers*. Each layer has *(i)* one regular residual block as in He et al. (2016), *(ii)* each stack has one stochastic layer at the end. The stacks are followed by a final stochastic layer in the encoder. We do downsampling at most once per stack. We used two layers per stack. For the decoder the structure of the encoder is reversed and convolutions are replaced by transposed convolutions.

For CIFAR we downsample twice so that the last stochastic layer has feature maps of size 8x8. We use between 3 and 5 ResNet blocks per stack. We choose learning rate from $\{1 \times 10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}\}$, the FouST scaling parameter from $\{0.5, 0.8, 0.9\}$, and the uniform interval is scaled by a factor from $\{0.01, 0.05, 0.1\}$ For CIFAR-10, we use the Adam (Kingma & Ba, 2017) optimizer and an discretized logistic mixture output model (Salimans et al., 2017) with 10 mixture components.

For mini-ImageNet we downsample thrice. We optimize using SGD and choose the learning rate from $\{2 \times 10^{-7}, 3 \times 10^{-7}, 4 \times 10^{-7}, 5 \times 10^{-7}\}$. We use a Gaussian output for this model with learned mean and variance.

# References

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv: 1412.6980.

Margulis, G. A. Probabilistic characteristics of graphs with large connectivity. *Problemy peredachi informatsii*, 10(2):101–108, 1974.

O'Donnell, R. *Analysis of boolean functions.* Cambridge University Press, 2014.

Russo, L. An approximate zero-one law. *Probability Theory and Related Fields*, 61(1):129–139, 1982.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. *arXiv:1701.05517 [cs, stat]*, January 2017. URL http://arxiv.org/abs/1701.05517. arXiv: 1701.05517.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pp. 3630–3638, 2016.