Attention for Inference Compilation

William Harvey*, Andreas Munk*, Atılım Güneş Baydin, Alexander Bergholm, Frank Wood

wsgh@cs.ubc.ca

*equal contribution

We present an improved neural architecture for amortized inference in probabilistic programs with complex control flow.

Inference compilation (Le et al., 2016, arXiv:1610.09900) learns ϕ so that $q(\mathbf{x}|\mathbf{y};\phi)$ is an amortized approximation of the posterior $p(\mathbf{x}|\mathbf{y})$ by minimizing the loss

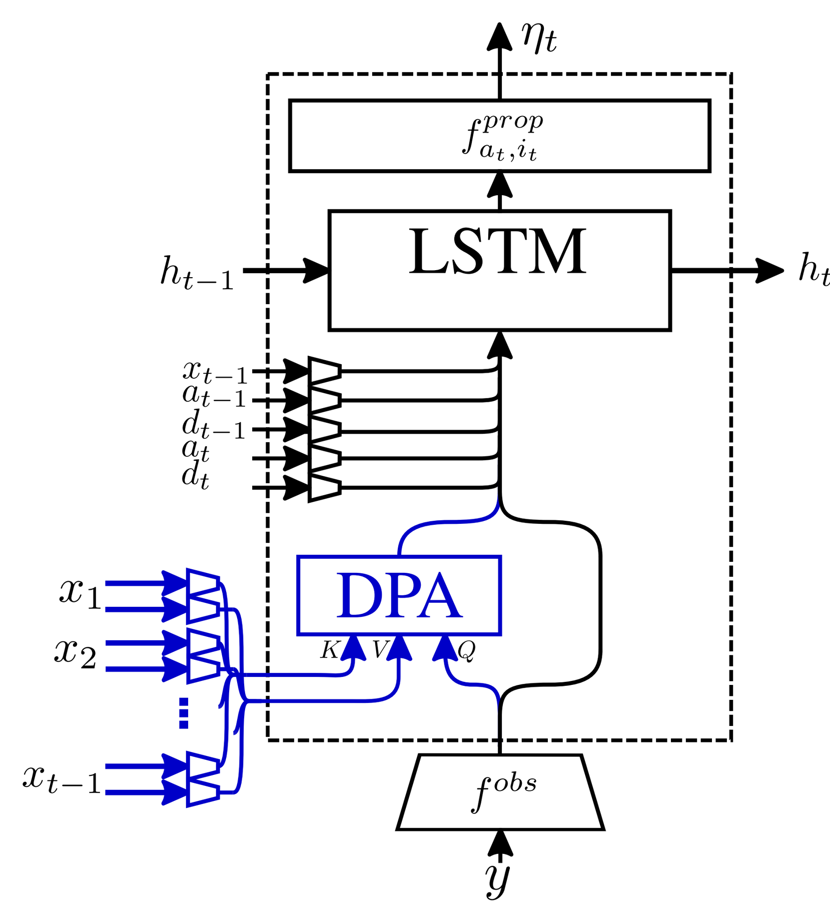
$$\mathbb{E}_{p(\mathbf{y})}[D_{\mathrm{KL}}(p(\mathbf{x}|\mathbf{y})||q(\mathbf{x}|\mathbf{y};\phi))] = \mathbb{E}_{p(\mathbf{x},\mathbf{y})}[-\log q(\mathbf{x}|\mathbf{y},\phi)] + \text{const}$$

$q(\mathbf{x}|\mathbf{y};\phi)$ is run alongside the probabilistic program, so can be used with complex programs without needing to learn the control flow. However, it must learn the dependencies between latent variables. We show that prior work can fail to do this.

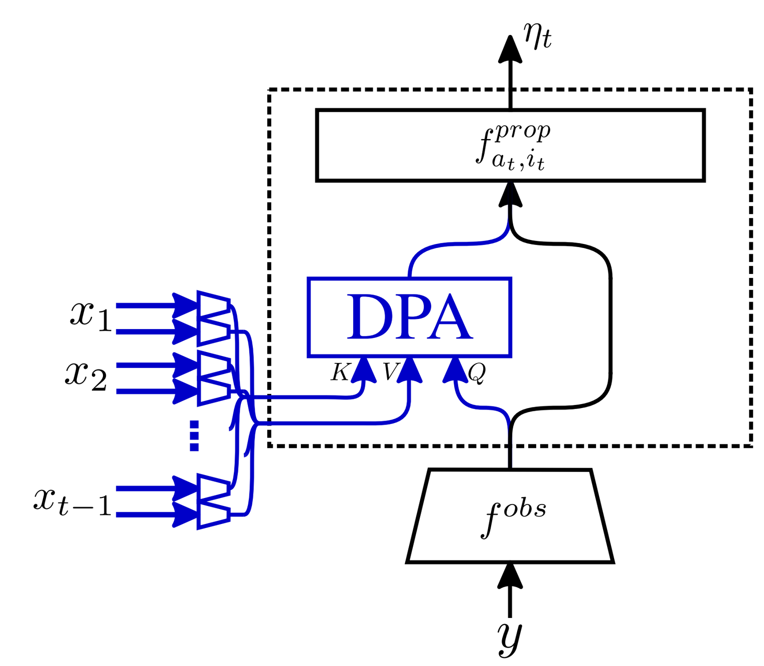
Attention has become an integral part of sequence modelling in fields such as NLP. We demonstrate that the transformer module (Vaswani et al., 2017, arXiv:1706.03762) improves the modelling of sequences of latent variables in the posterior.

The inference network runs alongside the probabilistic program, so does not need to learn the control flow. At each `sample` statement, the inference network outputs a proposal distribution conditioned on the observations and previous latent variables.

LSTM version:

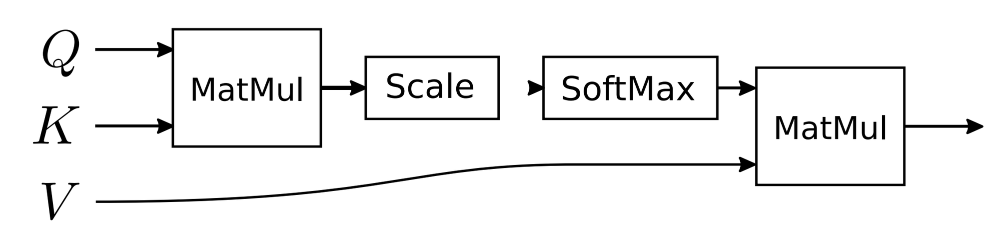


Feedforward (FF) version:



- y - observed variables
- η_t - parameters of proposal distribution for tth variable
- x_t - sampled value for tth variable
- a_t - address of tth variable
- i_t - number of times a_t has been encountered
- d_t - distribution type of tth variable in program
- h_t - hidden state at t

DPA (transformer) module:

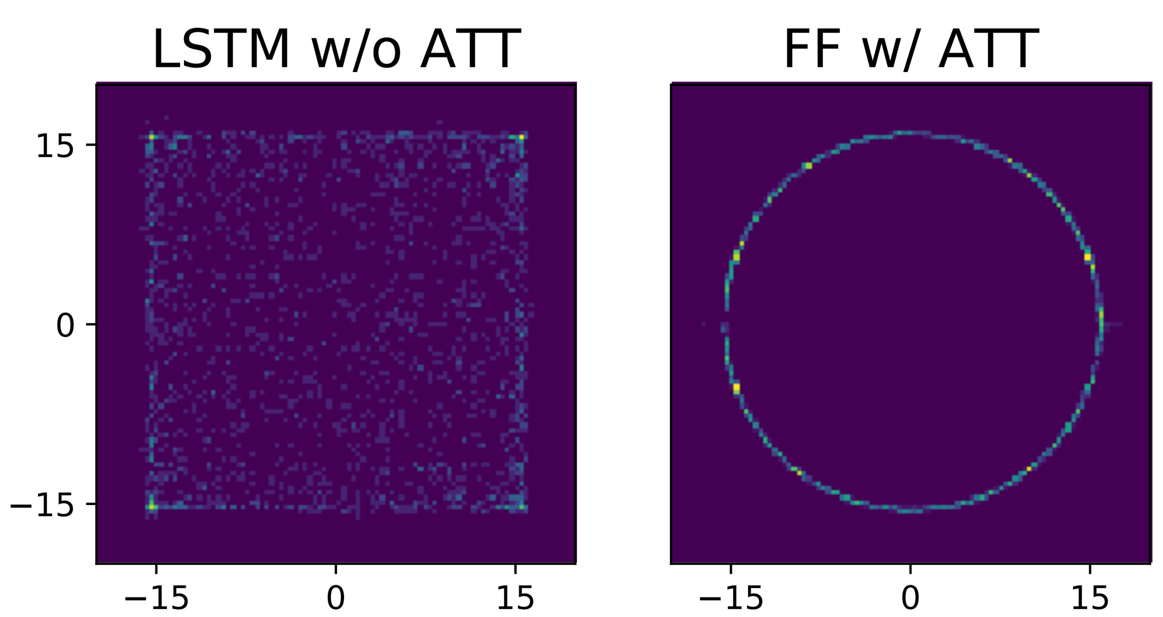


Our additions are in blue.

Pedagogical example: Find posterior over x and y given a noisy observation of $x^2 + y^2$; the true posterior has circular symmetry. Is this learned if nuisance variables are sampled between x and y?

```
x = sample(Normal(0, 10))
for _ in range(M):
    _ = sample(Normal(0,10))
y = sample(Normal(0,10))
observe(obs, Normal(x^2 + y^2, 0.1))
```

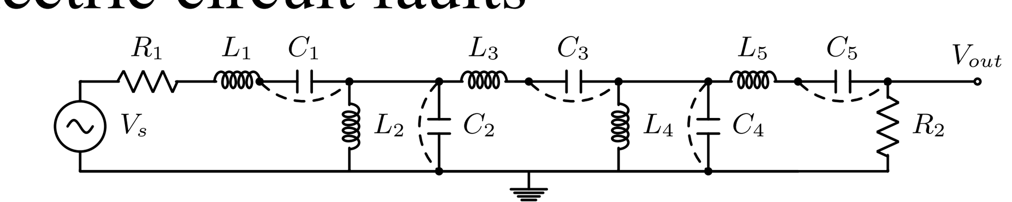
Proposal for (x, y) with $M = 50$, obs $= 15$



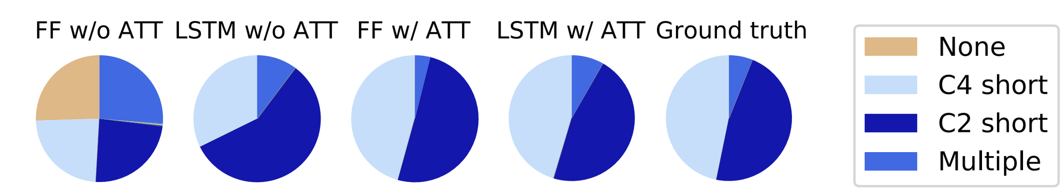
LSTM works with a few nuisance variables, but attention is required with $M \geq 50$.

We demonstrate better or equivalent performance on various datasets.

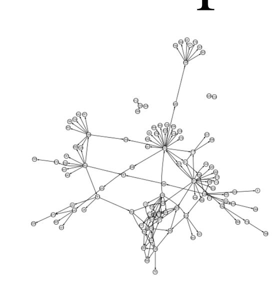
Electric circuit faults



Model of possible faults in an electric circuit. We observe the output voltage at 40 frequencies, and perform inference over the $\approx$ 40 latent variables corresponding to possible faults. For a fixed observation, the below diagram compares the proposal distribution from each network to the ground truth.

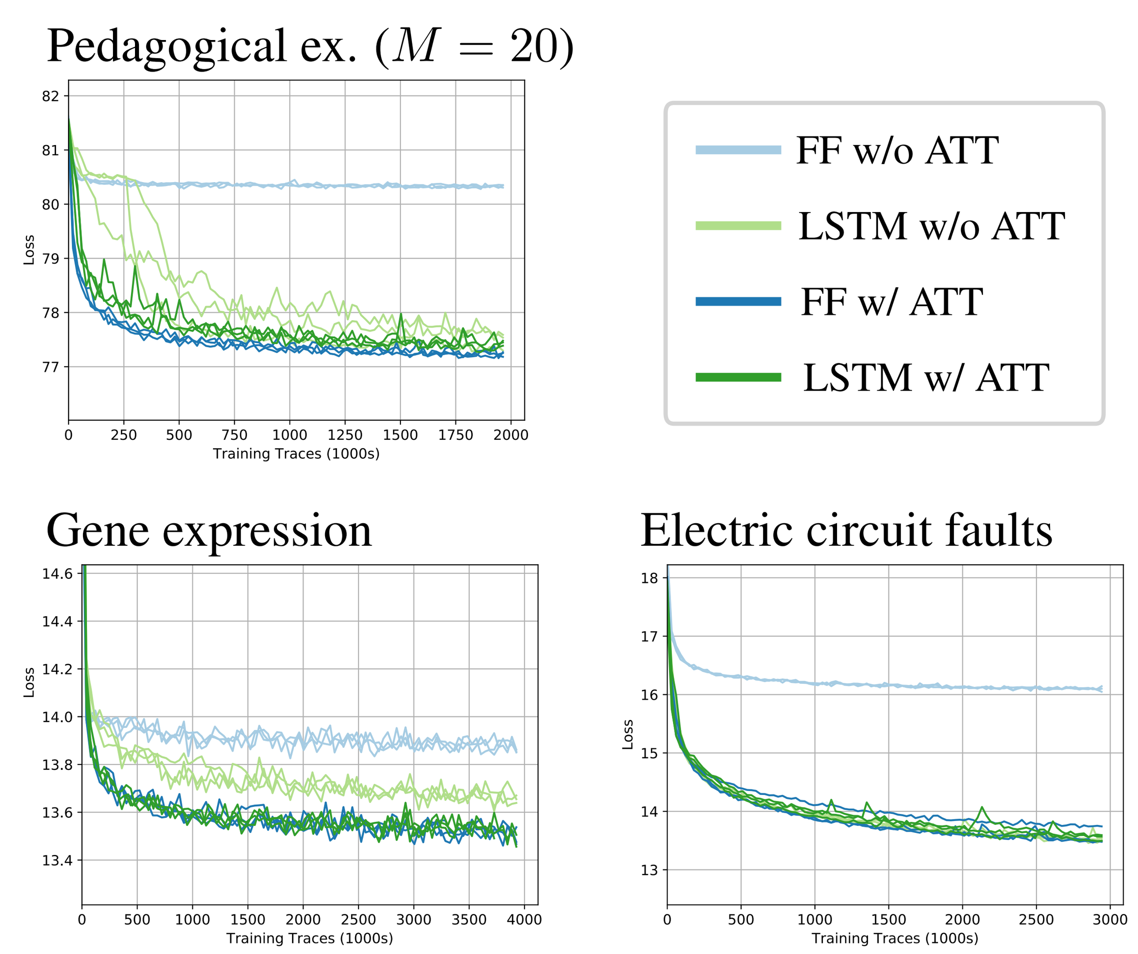


Gene expression



Publicly available model of plant gene expression, consisting of a 107 variable Bayesian network. We observe 40 leaf nodes, and infer the posterior over the rest.

Loss ($\mathbb{E}_{p(\mathbf{x},\mathbf{y})}[-\log q(\mathbf{x}|\mathbf{y},\phi)]$)



THE UNIVERSITY OF BRITISH COLUMBIA

UNIVERSITY OF OXFORD