# Supplementary material: Forward Amortized Inference for Likelihood-Free Variational Marginalization

## A: Hyperparameter optimization

In the reversed KL approach the joint-contrastive loss can be optimized simultaneously with respect to both $q$ and $p$. In the case of FAVI it is not possible to directly optimize the loss with respect to $p$ since $k(x)$ cannot be evaluated in closed-form. This problem can be overcome by rewriting the FAVI loss as an adversarial minimax problem using the log-density-ratio trick Tran et al. [2017]:

$$\begin{aligned} &D_{KL}(p(x,z) \| q(x,z)) = \\ &\mathbb{E}_{p(x)}\left[\log \frac{k(x)}{p(x)}\right] + \mathbb{E}_{p(x,z)}\left[\log \frac{q(z|x)}{p(z|x)}\right] \\ &= \mathbb{E}_{p(x)}[D_1^*(x)] + \mathbb{E}_{p(x,z)}[D_2^*(z|x)] \ , \end{aligned} \tag{1}$$

where $D_1^*(x)$ is the logit output of a nonparametric logistic regression trained to classify $k(x)$ samples from $p(x)$ samples:

$$\begin{aligned} &D_1^*(x) = \\ &\text{arginf}_D \left[\mathbb{E}_{p(x)}[\log \sigma(D_1(x))] - \mathbb{E}_{k(x)}[\log (1 - \sigma(D_1(x)))]\right] . \end{aligned} \tag{2}$$

Analogously, $D_2^*(x|z)$ is the logit output of a conditional (nonparametric) logistic regression trained to classify $q(z|x)$ samples from $p(z|x)$ samples:

$$\begin{aligned} D_2^*(x) = \text{arginf}_D &\left[\mathbb{E}_{q(z|x)p(x)}[\log \sigma(D_2(x))]\right. \\ &\left. - \mathbb{E}_{p(z|x)p(x)}[\log (1 - \sigma(D_2(x)))]\right] . \end{aligned} \tag{3}$$

In practice, the FAVI is approximated by restricting $D(x)$ to be a parametrized function such as a deep network. The gradient of the resulting loss is given by:

$$\nabla_p D_{KL}(q\|p) = \nabla_p \mathbb{E}_{p(x)}[D_1^*(x)] + \nabla_p \mathbb{E}_{p(x,z)}[D_2^*(z|x)] \ ,$$

which can be optimized by back-propagating through the $p(x)$ samples without requiring any direct evaluation of the density $p(x)$.

## B: Details of the network for variational Bayesian forecasting

The variational predictive distribution was parametrized by a kernel mexture network. The weights of the output kernel were given by the output of a deep network comprised 6 of dilated convolution layers with 30 one-dimensional kernels of length three and rectified linear units. The initial dilation factor was 1 and it was doubled after each layer. The wights were obtained from the activations of the last convolutinal layer by applying a fully connected layer. We initialized the bias terms to zero and the weights to samples drawn from a scaled Gaussian distribution. The resulting Bayesian variational forecaster was trained for 5000 iterations using the adaptive optimizer Adam. At each iteration a batch of 100 simulated training pairs was generated by integrating the Lorentz dynamical system. This procedure assures that the gradient of the network is unbiased since the batches are sampled from the real distribution.

## C: Details of the meta-learning network

The diagram of the architecture of our meta-classifier is shown in Fig. 1. In order to be able to adapt to different classification problems as specified by the different models in the ensemble, the network needs to receive feedback concerning the labels of the previous data points. This feedback is encoded as a vector of length $P \times C$, where $P$ is the number of predictors and $C$ is the numbers of classification classes. The feedback vector of the $n$-th sample takes the following form: $\boldsymbol{f^{n+1}} = (y_1^n \boldsymbol{x}^n \ , \ ... \ , \ y_C^n \boldsymbol{x}^n)$, where $y_c^n$ is the $c$-th component of the one-hot encoded label vector of the $n$-th sample. This feedback vector is fed to a layer of LSTM units through a dense linear map. The output of this first LSTM layer is fed via a dense linear map to another smaller intermediate LSTM layer. The outputs of these two LSTM layers are concatenated together with the current vector of predictors $\boldsymbol{x}^{n+1}$ and fed into a non-recurrent hidden layer via a dense linear map followed by an entry-wise Swish activation function. Finally, the probability vector of the current label is obtained with a softmax output layer.

## D: Details of the meta-learning ensemble

The performance of our approximate Bayes classifier on real-world data relies on the choice of the ensemble of models. Our aim is to define an ensemble prior that is appropriate for weakly structured classification tasks

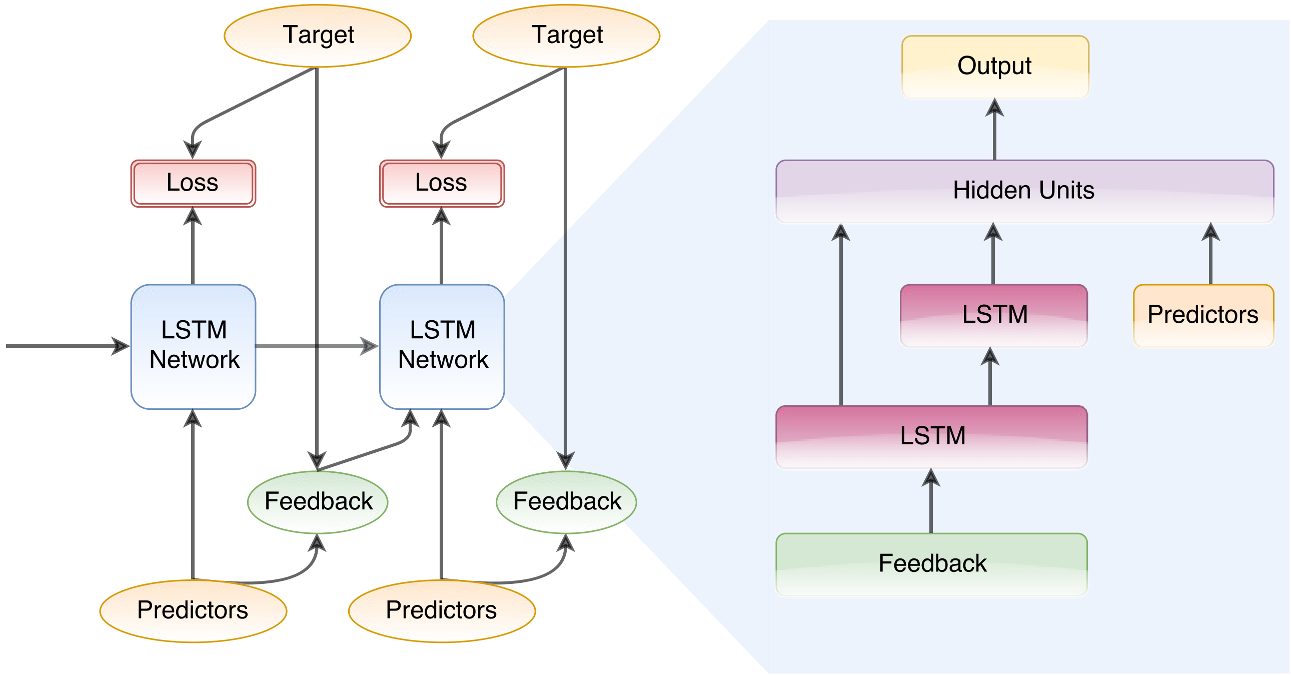

Figure 1: **A.** Architecture of the recurrent meta-classifier.

where we do not have any prior information about the structure of the predictors and of the statistical relationship between predictors and class labels. We used an ensemble of decision trees as methods based on these models tend to have the highest performance in these settings. We further increased the flexibility of the ensemble by using differentiable probabilistic decision trees. In these models, each split is determined by the inner product of the predictor with a vector of weights: $\boldsymbol{w}^\top \boldsymbol{x}$. This allows for diagonal decision rules at each split. Instead of using a deterministic decision rule we send the data to the left branch with probability $\sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$ and to the right branch with probability $1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$. Each leaf node was labeled with a random label. The depth of the tree was randomly sampled from 1 to 15. The weights and the biases at each node were sampled from a spherical normal distribution with mean 0 and variance 1. In order to further increase the flexibility of our ensemble we used a second family of classification models alongside the differentiable trees. In this second family, the classification dataset was generated as a set of samples normally distributed along the vertices of a 10-dimensional hypercube. Each class then comprised the samples associated with half of the vertices of the hypercube. These datasets were created using Scikit-learn toolbox (v0.18).

## References

D. Tran, R. Ranganath, and David M. Blei. Hierarchical implicit models and likelihood-free variational inference. *arXiv preprint arXiv:1702.08896*, 2017.