A. Derivation of the Lower Bound

As stated in the paper, the ELBO objective can be decomposed into a lower bound and an expected KL term, i.e.,

$$\begin{aligned} J(\boldsymbol{\theta}) = \int_{\mathbf{x}} \sum_{o} p(\boldsymbol{x}, o) \big(R(\boldsymbol{x}) - \log p(\boldsymbol{x}, o) \\ + \log \tilde{p}(o|\boldsymbol{x}) \big) d\boldsymbol{x} + \int_{\mathbf{x}} p(\boldsymbol{x}) \sum_{o} p(o|\boldsymbol{x}) \log \frac{p(o|\boldsymbol{x})}{\tilde{p}(o|\boldsymbol{x})}. \end{aligned} \tag{9}$$

We can verify that this decomposition is valid by using the identity $\log p(o|\boldsymbol{x}) = \log p(\boldsymbol{x}, o) - \log p(\boldsymbol{x})$, i.e.,

$$\begin{aligned} J(\boldsymbol{\theta}) &= \int_{\mathbf{x}} \sum_{o} p(\boldsymbol{x}, o) \big(R(\boldsymbol{x}) - \log p(\boldsymbol{x}, o) \\ &\quad + \log \tilde{p}(o|\boldsymbol{x}) \big) d\boldsymbol{x} \\ &\quad + \int_{\mathbf{x}} \sum_{o} p(\boldsymbol{x}) p(o|\boldsymbol{x}) \big(\log p(\boldsymbol{x}, o) - \log p(\boldsymbol{x}) \\ &\quad - \log \tilde{p}(o|\boldsymbol{x}) \big) d\boldsymbol{x}. \\ &= \int_{\mathbf{x}} \sum_{o} p(\boldsymbol{x}, o) \big(R(\boldsymbol{x}) - \log p(\boldsymbol{x}) \big) d\boldsymbol{x} \\ &= \int_{\mathbf{x}} p(\boldsymbol{x}) \big(R(\boldsymbol{x}) - \log p(\boldsymbol{x}) \big) d\boldsymbol{x}. \end{aligned} \tag{10}$$

We can see that Eq. 10 corresponds to the original definition of $L(\boldsymbol{\theta})$ in the paper.

B. Computation of the MMD

The Maximum Mean Discrepancy (Gretton et al., 2012) is a nonparametric divergence between mean embeddings in a Reproducible Kernel Hilbert Space. We approximate the MMD between two sample sets $\mathbf{X}$ and $\mathbf{Y}$ as

$$\begin{aligned} \mathrm{MMD}(\mathbf{X}, \mathbf{Y}) = \frac{1}{m^2} \sum_{i,j}^{m} k(x_i, x_j) + \frac{1}{n^2} \sum_{i,j}^{n} k(y_i, y_j) \\ - \frac{2}{mn} \sum_{i}^{m} \sum_{j}^{n} k(x_i, y_i). \end{aligned}$$

We use a squared exponential kernel given by

$$k(\mathbf{x}, \mathbf{y}) = \exp \left(-\frac{1}{\alpha} (\mathbf{x} - \mathbf{y})^\top \boldsymbol{\Sigma} (\mathbf{x} - \mathbf{y}) \right),$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix where each entry is set to the median of squared distances within the ground-truth set and the bandwidth α is chosen depending on the problem. When ground-truth samples are not available, we apply GESS (Nishihara et al., 2014) with large values for burn-in, thinning and chain lengths to produce baseline samples that are regarded as ground-truth. Note that obtaining these ground-truth samples is computationally very expensive, taking up to 2 days of computation time on 120 CPU cores. We estimate the MMD based on ten thousand ground-truth samples and two thousand samples from the given sampling method. For MCMC methods, we choose the two thousand most promising samples by applying a sufficient amount of burn-in and using the largest thinning that keeps at least two thousand samples in the set.

C. Component Optimization

As the Lagrangian of the optimization problem for the component update corresponds to the Lagrangian of MORE (Abdolmaleki et al., 2016) with $\omega = 1$, the solution has the form

$$p(\boldsymbol{x}|o) \propto q(\boldsymbol{x}|o)^{\frac{\eta}{\eta+1}} \exp\left(\tilde{r}_o(\boldsymbol{x})\right)^{\frac{1}{\eta+1}}, \tag{11}$$

where we substituted $\omega = 1$ in Equation 5.

When the quadratic reward surrogate is given as

$$\tilde{r}_o(\boldsymbol{x}) = -\frac{1}{2} \boldsymbol{x}^\top \boldsymbol{R} \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{r},$$

the parameters $\boldsymbol{R}$ and $\boldsymbol{r}$ (which are learned with weighted least squares) correspond to the natural parameters of a multivariate normal distribution

$$p_r(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_r = \boldsymbol{R}^{-1}\boldsymbol{r}, \boldsymbol{\Sigma}_r = \boldsymbol{R}^{-1}) \propto \exp\left(\tilde{r}_o(\boldsymbol{x})\right).$$

Hence, the log-densities of $p(\boldsymbol{x}|o)$ are given by a linear interpolation of the log-densities of $q(\boldsymbol{x}|o)$ and $p_r(\boldsymbol{x})$, i.e.

$$\log p(\boldsymbol{x}|o) = \frac{\eta}{\eta+1} \log q(\boldsymbol{x}|o) + \frac{1}{\eta+1} \log p_r(\boldsymbol{x}) + \text{const.}$$

The natural parameters of $p(\boldsymbol{x}|o)$ are therefore given by

$$\boldsymbol{P} = \frac{1}{\eta+1} (\eta \boldsymbol{Q} + \boldsymbol{R}), \qquad \boldsymbol{p} = \frac{1}{\eta+1} (\eta \boldsymbol{q} + \boldsymbol{r}),$$

where $\boldsymbol{Q} = \boldsymbol{\Sigma}_q^{-1}$ and $\boldsymbol{q} = \boldsymbol{\Sigma}_q^{-1}\boldsymbol{\mu}_q$ are the natural parameters of $q(\boldsymbol{x}|o)$.

As a function of the Lagrangian multiplier η,

$$p(\boldsymbol{x}|o, \eta) = \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}_p = \boldsymbol{P}(\eta)^{-1}\boldsymbol{p}(\eta), \boldsymbol{\Sigma}_p = \boldsymbol{P}(\eta)^{-1}\right)$$

defines an e-geodesic, i.e. a straight line connecting $q(\boldsymbol{x}|o)$ and $p_r(\boldsymbol{x})$ in logarithmic scale. During optimization we want to find the largest *step-size* η such that $p(\boldsymbol{x}|o, \eta)$ stays within the trust region. As we are minimizing a scalar on a convex function, a simple line-search would be feasible. However, the dual objective

$$G_o(\eta) = \eta\epsilon(o) + \eta \log Z(\boldsymbol{Q}, \boldsymbol{q}) - (\eta+1) \log Z(\boldsymbol{P}, \boldsymbol{p}),$$

where $\log Z(\boldsymbol{X}, \boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}^\top \boldsymbol{X}^{-1}\boldsymbol{x} + \log|2\pi\boldsymbol{X}^{-1}|)$ is the log partition function of a Gaussian with natural parameters $\boldsymbol{X}$ and $\boldsymbol{x}$, as well as the gradient

$$\frac{dG_o(\eta)}{d\eta} = \epsilon(o) - \mathrm{KL}(p(\boldsymbol{x}|o,\eta)||q(\boldsymbol{x}|o))$$

can be computed with little overhead and hence we use L-BFGS for dual descent.

D. Weight Optimization

The optimization of the distribution over weights is similar to the optimization of the components but we are optimizing over a discrete distribution rather than a multivariate normal. Similar to the component optimization, the optimal distribution has the form

$$p(o) \propto q(o)^{\frac{\eta_w}{\eta_w+1}} \exp\left(\tilde{r}_w(o)\right)^{\frac{1}{\eta_w+1}}, \qquad (12)$$

and corresponds to a log-linear interpolation between the last distribution $q(o)$ and a distribution $p_r(o) \propto \exp(\tilde{r}_w(o))$ that is specified by the reward function. The optimal stepsize η_w can be found by minimizing the dual

$$G_w(\eta_w) = \eta_w\epsilon_w + (1+\eta_w)\log\sum_o p(o|\eta_w)$$

based on the gradient

$$\frac{dG_w(\eta_w)}{d\eta_w} = \epsilon - \mathrm{KL}(p(o|\eta)||q(o))$$

with L-BFGS.

E. Hyper-parameters

Table 1 lists the hyper-parameters as well as their values for the experiments. We will now briefly discuss some of these hyper-parameters.

E.1. KL bounds

The trust regions are necessary for the component updates in order to ensure that the components stay within regions where their local reward surrogate $\tilde{r}_o(\boldsymbol{x})$ remains valid. As the reward surrogate is updated in each EM iteration, we also update the reference distribution $q(\boldsymbol{x}|o)$ after each EM iteration. However, this may allow the component to enter regions that are insufficiently covered by samples after several EM iterations which would result in bad local surrogates. We therefore compute the KL bound based on the effective number of samples within the active set, namely the KL bound is given by

$$\epsilon(o) = \min(1\mathrm{e}{-3}, 1\mathrm{e}{-5} \cdot n_{\text{eff}}(o)),$$

where the effective sample size is computed based on the importance weights

$$n_{\text{eff}}(o) = \frac{\left(\sum_{i=1}^{N_s} w_i(o)\right)^2}{\sum_{i=1}^{N_s} w_i(o)^2}.$$

Table 1: A list of the hyper-parameters of VIPS as well their values used during the experiments.

DESCRIPTION	VALUE
MAXIMUM NUMBER OF COMPONENTS	$1, 5, 40$
NUMBER OF EM ITERATIONS	10
KL BOUND FOR WEIGHTS	1e−2
MAXIMUM KL BOUND FOR COMPONENTS	1e−3
KL BOUND FACTOR FOR COMPONENTS	1e−5
NUMBER OF SAMPLES PER COMPONENT	$10 \cdot D$
NUMBER OF INITIAL SAMPLES	$20, 20000$
SAMPLE REUSE FACTOR	3
ADDING RATE FOR COMPONENTS	30
DELETION RATE FOR COMPONENTS	300
MINIMUM WEIGHT	1e−7
ℓ_2-REGULARIZATION FOR WLS	1e−10

As we ignore the weights for sampling during training, the KL bound for the weights is not critical and could even be dropped. However, for the experiments we chose a KL bound of $\epsilon_w = 1\mathrm{e}{-2}$, because it seems sensible to prevent large jumps in the log responsibilities.

E.2. Samples

As stated in the paper, we draw $10D$ samples *per component* and roughly reuse the samples from the last 3 most recent iterations. For the experiments, we drew 20000 additional samples from the initial mixture at the beginning of the optimization for better initial exploration. However, we lowered this value to 20 for VIPS1 which often already converged after 20000 iterations.

E.3. Adding and Deleting Components

We added a single new component every third sampling iteration and initialized its weight to $1\mathrm{e}{-7}$. For computing the score e_i for deciding where to add the component, we use $\gamma = 500$. This hyper-parameter is probably the least intuitive to be chosen. When γ is too small, new modes may only be discovered when we have sampled close to their peak. However, when γ is too large we might add components at irrelevant regions, especially when the target distribution has heavy tales. However, we found $\gamma = 500$ to produce good results among all our experiments.

We delete a component when its weight was below $1\mathrm{e}{-7}$ for the last 300 EM-Iterations (i.e. 30 sampling iterations). We do not want to keep components with lower weight, because their effect on the approximation would be marginal.

References

Abdolmaleki, A., Lioutikov, R., Lua, N., Reis, L. P., Peters, J., and Neumann, G. Model-Based Relative Entropy Stochastic Search. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 153–154, 2016

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A Kernel Two-sample Test. *Journal of Machine Learning Research*, 13:723–773, March 2012. ISSN 1532-4435

Nishihara, R., Murray, I., and Adams, R. P. Parallel MCMC with Generalized Elliptical Slice Sampling. *Journal of Machine Learning Research*, 15(1):2087–2112, January 2014