Kernel-Based Just-In-Time Learning for Passing Expectation Propagation Messages
SUPPLEMENTARY MATERIAL

A MEDIAN HEURISTIC FOR GAUSSIAN KERNEL ON MEAN EMBEDDINGS

In the proposed KJIT, there are two kernels: the inner kernel k for computing mean embeddings, and the outer Gaussian kernel κ defined on the mean embeddings. Both of the kernels depend on a number of parameters. In this section, we describe a heuristic to choose the kernel parameters. We emphasize that this heuristic is merely for computational convenience. A full parameter selection procedure like cross validation or evidence maximization will likely yield a better set of parameters. We use this heuristic in the initial mini-batch phase before the actual online learning.

Let $\{r_i^{(l)} \mid l = 1, \ldots, c, \text{ and } i = 1, \ldots, n\}$ be a set of n incoming message tuples collected during the mini-batch phase, from c variables neighboring the factor. Let $R_i := (r_i^{(l)})_{l=1}^c$ be the i^{th} tuple, and let $\mathsf{r}_i := \times_{l=1}^c r_i^{(l)}$ be the product of incoming messages in the i^{th} tuple. Define S_i and s_i to be the corresponding quantities of another tuple of messages. We will drop the subscript i when considering only one tuple.

Recall that the kernel on two tuples of messages R and S is given by

$$\begin{aligned}\kappa(R, S) = \kappa(\mathsf{r}, \mathsf{s}) &= \exp\left(-\frac{\|\mu_\mathsf{r} - \mu_\mathsf{s}\|_\mathcal{H}^2}{2\gamma^2}\right) \\ &= \exp\left(-\frac{1}{2\gamma^2}\langle\mu_\mathsf{r}, \mu_\mathsf{r}\rangle + \frac{1}{\gamma^2}\langle\mu_\mathsf{r}, \mu_\mathsf{s}\rangle - \frac{1}{2\gamma^2}\langle\mu_\mathsf{s}, \mu_\mathsf{s}\rangle\right),\end{aligned}$$

where $\langle\mu_\mathsf{r}, \mu_\mathsf{s}\rangle = \mathbb{E}_{x\sim\mathsf{r}}\mathbb{E}_{y\sim\mathsf{s}}k(x-y)$. The inner kernel k is a Gaussian kernel defined on the domain $\mathcal{X} := \mathcal{X}^{(1)} \times \cdots \times \mathcal{X}^{(c)}$ where $\mathcal{X}^{(l)}$ denotes the domain of $r^(l)$. For simplicity, we assume that $\mathcal{X}^{(l)}$ is one-dimensional. The Gaussian kernel k takes the form

$$k(x - y) = \exp\left(-\frac{1}{2}(x - y)^\top \Sigma^{-1}(x - y)\right) = \prod_{l=1}^{c} \exp\left(-\frac{(x_j - y_j)^2}{2\sigma_l^2}\right),$$

where $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_c^2)$. The heuristic for choosing $\sigma_1^2, \ldots, \sigma_c^2$ and γ is as follows.

1. Set $\sigma_l^2 := \frac{1}{n}\sum_{i=1}^n \mathbb{V}_{x_l\sim r_i^{(l)}}[x_l]$ where $\mathbb{V}_{x_l\sim r_i^{(l)}}[x_l]$ denotes the variance of $r_i^{(l)}$.
2. With $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_c^2)$ as defined in the previous step, set $\gamma^2 := \mathrm{median}\left(\{\|\mu_{\mathsf{r}_i} - \mu_{\mathsf{s}_j}\|^2\}_{i,j=1}^n\right)$.

B KERNELS AND RANDOM FEATURES

This section reviews relevant kernels and their random feature representations.

B.1 RANDOM FEATURES

This section contains a summary of Rahimi and Recht (2007)'s random Fourier features for a translation invariant kernel.

A kernel $k(x, y) = \langle\phi(x), \phi(y)\rangle$ in general may correspond to an inner product in an infinite-dimensional space whose feature map ϕ cannot be explicitly computed. In Rahimi and Recht (2007), methods of computing an approximate feature maps $\hat{\phi}$ were proposed. The approximate feature maps are such that $k(x, y) \approx \hat{\phi}(x)^\top\hat{\phi}(y)$ (with equality in expectation) where $\hat{\phi}(x), \hat{\phi}(y) \in \mathbb{R}^D$ and D is the number of random features. High D yields a better approximation with higher computational cost. Assume $k(x, y) = k(x-y)$ (translation invariant) and $x, y \in \mathbb{R}^d$. Random Fourier features $\hat{\phi}(x) \in \mathbb{R}^D$ such that $k(x, y) \approx \hat{\phi}(x)^\top\hat{\phi}(y)$ are generated as follows:

1. Compute the Fourier transform $\hat{k}$ of the kernel k: $\hat{k}(\omega) = \frac{1}{2\pi}\int e^{-j\omega^\top\delta}k(\delta)\, d\delta$.
2. Draw D i.i.d. samples $\omega_1, \ldots, \omega_D \in \mathbb{R}^d$ from $\hat{k}$.
3. Draw D i.i.d samples $b_1, \ldots, b_D \in \mathbb{R}$ from $U[0, 2\pi]$ (uniform distribution).
4. $\hat{\phi}(x) = \sqrt{\frac{2}{D}}\left(\cos\left(\omega_1^\top x + b_1\right), \ldots, \cos\left(\omega_D^\top x + b_D\right)\right)^\top \in \mathbb{R}^D$

Why It Works

Theorem 1. *Bochner's theorem (Rudin, 2013). A continuous kernel $k(x, y) = k(x - y)$ on $\mathbb{R}^m$ is positive definite iff $k(\delta)$ is the Fourier transform of a non-negative measure.*

Furthermore, if a translation invariant kernel $k(\delta)$ is properly scaled, Bochner's theorem guarantees that its Fourier transform $p(\omega)$ is a probability distribution. From this fact, we have

$$k(x-y) = \int \hat{k}(\omega) e^{j\omega^\top (x-y)}\, d\omega = \mathbb{E}_\omega \left[\eta_\omega(x)\eta_\omega(y)^*\right],$$

where $j = \sqrt{-1}$, $\eta_\omega(x) = e^{j\omega^\top x}$ and $\cdot^*$ denotes the complex conjugate. Since both $\hat{k}$ and k are real, the complex exponential contains only the cosine terms. Drawing D samples lowers the variance of the approximation.

Theorem 2. *Separation of variables. Let $\hat{f}$ be the Fourier transform of f. If $f(x_1, \ldots, x_d) = f_1(x_1) \cdots f_d(x_d)$, then $\hat{f}(\omega_1, \ldots, \omega_d) = \prod_{i=1}^d \hat{f}_i(\omega_i)$.*

Theorem 2 suggests that the random Fourier features can be extended to a product kernel by drawing ω independently for each kernel.

B.2 MV (MEAN-VARIANCE) KERNEL

Assume there are c incoming messages $R := \left(r^{(l)}\right)_{l=1}^c$ and $S := \left(s^{(l)}\right)_{l=1}^c$. Assume that

$$\begin{aligned}
\mathbb{E}_{r^{(l)}}[x] &= m_l \\
\mathbb{V}_{r^{(l)}}[x] &= v_l \\
\mathbb{E}_{s^{(l)}}[y] &= \mu_l \\
\mathbb{V}_{s^{(l)}}[y] &= \sigma_l^2.
\end{aligned}$$

Incoming messages are not necessarily Gaussian. The MV (mean-variance) kernel is defined as a product kernel on means and variances.

$$\kappa_{\text{mv}}(R, S) = \prod_{i=1}^c k\left(\left(m_i - \mu_i\right)/w_i^m\right) \prod_{i=1}^c k\left(\left(v_i - \sigma_i^2\right)/w_i^v\right),$$

where k is a Gaussian kernel with unit width. The kernel κ_{mv} has $P := (w_1^m, \ldots, w_c^m, w_1^v, \ldots, w_c^v)$ as its parameters. With this kernel, we treat messages as finite dimensional vectors. All incoming messages $(s^{(i)})_{i=1}^c$ are represented as $\left(\mu_1, \ldots, \mu_c, \sigma_1^2, \ldots, \sigma_c^2\right)^\top$. This treatment reduces the problem of having distributions as inputs to the familiar problem of having input points from a Euclidean space. The random features of Rahimi and Recht (2007) can be applied straightforwardly.

B.3 EXPECTED PRODUCT KERNEL

Given two distributions $r(x) = \mathcal{N}(x; m_r, V_r)$ and $s(y) = \mathcal{N}(y; m_s, V_s)$ (d-dimensional Gaussian), the expected product kernel is defined as

$$\kappa_{\text{pro}}(r, s) = \langle \mu_r, \mu_s \rangle_{\mathcal{H}} = \mathbb{E}_r \mathbb{E}_s k(x - y),$$

where $\mu_r := \mathbb{E}_r k(x, \cdot)$ is the mean embedding of r, and we assume that the kernel k associated with $\mathcal{H}$ is translation invariant i.e., $k(x, y) = k(x - y)$. The goal here is to derive random Fourier features for the expected product kernel. That is, we aim to find $\hat{\phi}$ such that $\kappa_{\text{pro}}(r, s) \approx \hat{\phi}(r)^\top \hat{\phi}(s)$ and $\hat{\phi} \in \mathbb{R}^D$.

We first give some results which will be used to derive the Fourier features for inner product of mean embeddings.

Lemma 3. *If $b \sim \mathcal{N}(b; 0, \sigma^2)$, then $\mathbb{E}[\cos(b)] = \exp\left(-\frac{1}{2}\sigma^2\right)$.*

Proof. We can see this by considering the characteristic function of $x \sim \mathcal{N}(x; \mu, \sigma^2)$ which is given by

$$c_x(t) = \mathbb{E}_x\left[\exp(itb)\right] = \exp\left(itm - \frac{1}{2}\sigma^2 t^2\right).$$

For $m = 0, t = 1$, we have

$$c_b(1) = \mathbb{E}_b\left[\exp(ib)\right] = \exp\left(-\frac{1}{2}\sigma^2\right) = \mathbb{E}_b\left[\cos(b)\right],$$

where the imaginary part $i\sin(tb)$ vanishes. □

From Rahimi and Recht (2007) which provides random features for $k(x-y)$, we immediately have

$$\begin{aligned}\mathbb{E}_r\mathbb{E}_s k(x-y) &\approx \mathbb{E}_r\mathbb{E}_s \frac{2}{D}\sum_{i=1}^{D}\cos\left(w_i^\top x + b_i\right)\cos\left(w_i^\top y + b_i\right)\\ &= \frac{2}{D}\sum_{i=1}^{D}\mathbb{E}_{r(x)}\cos\left(w_i^\top x + b_i\right)\mathbb{E}_{s(y)}\cos\left(w_i^\top y + b_i\right),\end{aligned}$$

where $\{w_i\}_{i=1}^D \sim \hat{k}(w)$ (Fourier transform of k) and $\{b_i\}_{i=1}^D \sim U\left[0, 2\pi\right]$.

Consider $\mathbb{E}_{r(x)}\cos\left(w_i^\top x + b_i\right)$. Define $z_i = w_i^\top x + b_i$. So $z_i \sim \mathcal{N}(z_i; w_i^\top m_r + b_i, w_i^\top V_r w_i)$. Let $d_i \sim \mathcal{N}(0, w_i^\top V_r w_i)$. Then, $r(d_i + w_i^\top m_r + b_i) = \mathcal{N}(w_i^\top m_r + b_i, w_i^\top V_r w_i)$ which is the same distribution as that of z_i. From these definitions we have,

$$\begin{aligned}\mathbb{E}_{r(x)}\cos\left(w_i^\top x + b_i\right) &= \mathbb{E}_{r(z_i)}\cos(z_i)\\ &= \mathbb{E}_{r(d_i)}\cos\left(d_i + w_i^\top m_r + b_i\right)\\ &\overset{(a)}{=} \mathbb{E}_{r(d_i)}\cos(d_i)\cos(w_i^\top m_r + b_i) - \mathbb{E}_{r(d_i)}\sin(d_i)\sin(w_i^\top m_r + b_i)\\ &\overset{(b)}{=} \cos(w_i^\top m_r + b_i)\mathbb{E}_{r(d_i)}\cos(d_i)\\ &\overset{(c)}{=} \cos(w_i^\top m_r + b_i)\exp\left(-\frac{1}{2}w_i^\top V_r w_i\right),\end{aligned}$$

where at (a) we use $\cos(\alpha+\beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$. We have (b) because $\sin$ is an odd function and $\mathbb{E}_{r(d_i)}\sin(d_i) = 0$. The last equality (c) follows from Lemma 3. It follows that the random features $\hat{\phi}(r) \in \mathbb{R}^D$ are given by

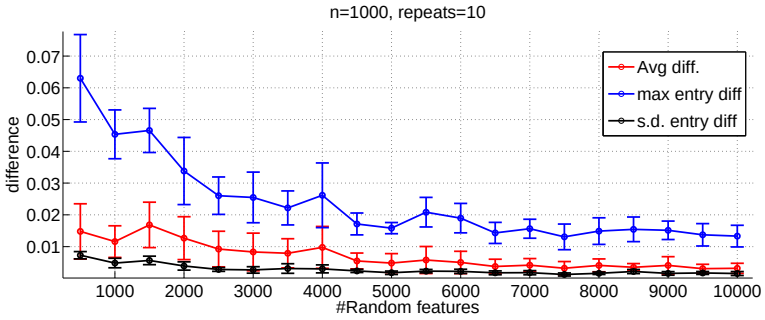
$$\hat{\phi}(r) = \sqrt{\frac{2}{D}}\begin{pmatrix}\cos(w_1^\top m_r + b_1)\exp\left(-\frac{1}{2}w_1^\top V_r w_1\right)\\ \vdots\\ \cos(w_D^\top m_r + b_D)\exp\left(-\frac{1}{2}w_D^\top V_r w_D\right)\end{pmatrix}.$$

Notice that the translation invariant kernel k provides $\hat{k}$ from which $\{w_i\}_i$ are drawn. For a different type of distribution r, we only need to be able to compute $\mathbb{E}_{r(x)}\cos\left(w_i^\top x + b_i\right)$. With $\hat{\phi}(r)$, we have $\kappa_{\text{pro}}(r,s) \approx \hat{\phi}(r)^\top\hat{\phi}(s)$ with equality in expectation.

Analytic Expression for Gaussian Case For reference, if r, s are normal distributions and k is a Gaussian kernel, an analytic expression is available. Assume $k(x-y) = \exp\left(-\frac{1}{2}(x-y)^\top\Sigma^{-1}(x-y)\right)$ where Σ is the kernel parameter. Then

$$\begin{aligned}\mathbb{E}_r\mathbb{E}_s k(x-y) &= \sqrt{\frac{\det(D_{rs})}{\det(\Sigma^{-1})}}\exp\left(-\frac{1}{2}(m_r - m_s)^\top D_{rs}(m_r - m_s)\right),\\ D_{rs} &:= (V_r + V_s + \Sigma)^{-1}.\end{aligned}$$

Approximation Quality The following result compares the randomly generated features to the true kernel matrix using various numbers of random features D. For each D, we repeat 10 trials, where the randomness in each trial arises from the construction of the random features. Samples are univariate normal distributions $\mathcal{N}(m, v)$ where $m \sim \mathcal{N}(0, 9)$ and $v \sim \text{Gamma}(3, 1/4)$ (shape-rate parameterization). The kernel parameter was $\Sigma := \sigma^2 I$ where $\sigma^2 = 3$.



"max entry diff" refers to the maximum entry-wise difference between the true kernel matrix and the approximated kernel matrix.

B.4 PRODUCT KERNEL ON MEAN EMBEDDINGS

Previously, we have defined an expected product kernel on single distributions. One way to define a kernel between two tuples of more than one incoming message is to take a product of the kernels defined on each message.

Let $\mu_{r^{(l)}} := \mathbb{E}_{r^{(l)}(a)} k^{(l)}(\cdot, a)$ be the mean embedding (Smola et al., 2007) of the distribution $r^{(l)}$ into RKHS $\mathcal{H}^{(l)}$ induced by the kernel k. Assume $k^{(l)} = k^{(l)}_{\text{gauss}}$ (Gaussian kernel) and assume there are c incoming messages $R := (r^{(i)}(a^{(i)}))_{i=1}^{c}$ and $S := (s^{(i)}(b^{(i)}))_{i=1}^{c}$. A product of expected product kernels is defined as

$$\begin{aligned}\kappa_{\text{pro, prod}}(R, S) &:= \left\langle \bigotimes_{l=1}^{c} \mu_{r^{(l)}}, \bigotimes_{l=1}^{c} \mu_{s^{(l)}} \right\rangle_{\otimes_l \mathcal{H}^{(l)}} \\ &= \prod_{l=1}^{c} \mathbb{E}_{r^{(l)}(a^{(l)})} \mathbb{E}_{s^{(l)}(b^{(l)})} k^{(l)}_{\text{gauss}}\left(a^{(l)}, b^{(l)}\right) \approx \hat{\phi}(R)^\top \hat{\phi}(S),\end{aligned}$$

where $\hat{\phi}(R)^\top \hat{\phi}(S) = \prod_{l=1}^{c} \hat{\phi}^{(l)}(r^{(l)})^\top \hat{\phi}^{(l)}(s^{(l)})$. The feature map $\hat{\phi}^{(l)}(r^{(l)})$ can be estimated by applying the random Fourier features to $k^{(l)}_{\text{gauss}}$ and taking the expectations $\mathbb{E}_{r^{(l)}(a)} \mathbb{E}_{s^{(l)}(b)}$. The final feature map is $\hat{\phi}(R) = \hat{\phi}^{(1)}(r^{(1)}) \circledast \hat{\phi}^{(2)}(r^{(2)}) \circledast \cdots \circledast \hat{\phi}^{(c)}(r^{(c)}) \in \mathbb{R}^{D^c}$, where $\circledast$ denotes a Kronecker product and we assume that $\hat{\phi}^{(l)} \in \mathbb{R}^D$ for $l \in \{1, \ldots, c\}$.

B.5 SUM KERNEL ON MEAN EMBEDDINGS

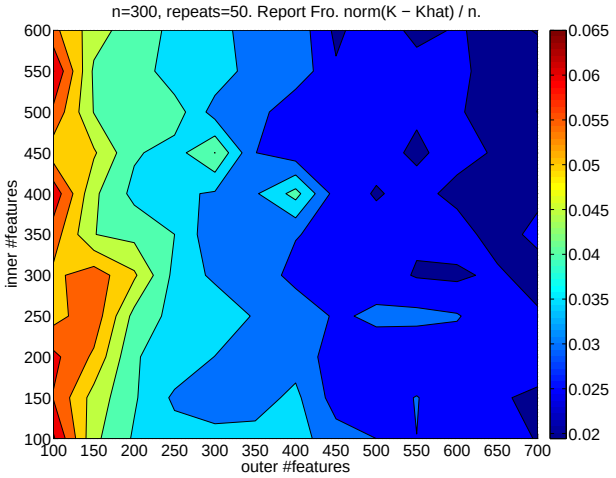
If we instead define the kernel as the sum of c kernels, we have

$$\begin{aligned}\kappa_{\text{pro, sum}}(R, S) &= \sum_{l=1}^{c} \langle \mu_{r^{(l)}}, \mu_{s^{(l)}} \rangle_{\mathcal{H}^{(l)}} \\ &\approx \sum_{l=1}^{c} \hat{\phi}^{(l)}(r^{(l)})^\top \hat{\phi}^{(l)}(s^{(l)}) \\ &= \hat{\varphi}(R)^\top \hat{\varphi}(S),\end{aligned}$$

where $\hat{\varphi}(R) := \left(\hat{\phi}^{(1)}(r^{(1)})^\top, \ldots, \hat{\phi}^{(c)}(r^{(c)})^\top\right)^\top \in \mathbb{R}^{cD}$.

B.6 NUMBER OF RANDOM FEATURES FOR GAUSSIAN KERNEL ON MEAN EMBEDDINGS

We quantify the effect of D_{in} and D_{out} empirically as follows. We generate 300 Gaussian messages, compute the true Gram matrix and the approximate Gram matrix given by the random features, and report the Frobenius norm of the difference of the two matrices on a grid of D_{in} and D_{out}. For each $(D_{\text{in}}, D_{\text{out}})$, we repeat 20 times with a different set of random features and report the averaged Frobenius norm.



The result suggests that D_{out} has more effect in improving the approximation.

C MORE DETAILS ON EXPERIMENT 1: BATCH LEARNING

There are a number of kernels on distributions we may use for just-in-time learning. To find the most suitable kernel, we compare the performance of each on a collection of incoming and output messages at the logistic factor in the binary logistic regression problem i.e., same problem as in experiment 1 in the main text. All messages are collected by running EP 20 times on generated toy data. Only messages in the first five iterations are considered as messages passed in the early phase of EP vary more than in a near-convergence phase. The regression output to be learned is the numerator of (1).

A training set of 5000 messages and a test set of 3000 messages are obtained by subsampling all the collected messages. Where random features are used, kernel widths and regularization parameters are chosen by leave-one-out cross validation. To get a good sense of the approximation error from the random features, we also compare with incomplete Cholesky factorization (denoted by IChol), a widely used Gram matrix approximation technique. We use hold-out cross validation with randomly chosen training and validation sets for parameter selection, and kernel ridge regression in its dual form when the incomplete Cholesky factorization is used.

Let f be the logistic factor and $m_{f\to i}$ be an outgoing message. Let $q_{f\to i}$ be the ground truth belief message (numerator) associated with $m_{f\to i}$. The error metric we use is $\mathrm{KL}[q_{f\to i}\,||\,\hat{q}_{f\to i}]$ where $\hat{q}_{f\to i}$ are the belief messages estimated by a learned regression function. The following table reports the mean of the log KL-divergence and standard deviations.

	mean log KL	**s.d. of log KL**
Random features + MV Kernel	-6.96	1.67
Random features + Expected product kernel on joint embeddings	-2.78	1.82
Random features + Sum of expected product kernels	-1.05	1.93
Random features + Product of expected product kernels	-2.64	1.65
Random features + Gaussian kernel on joint embeddings (KJIT)	-8.97	1.57
IChol + sum of Gaussian kernel on embeddings	-2.75	2.84
IChol + Gaussian kernel on joint embeddings	-8.71	1.69
Breiman's random forests (Breiman, 2001)	-8.69	1.79
Extremely randomized trees (Geurts et al., 2006)	-8.90	1.59
Eslami et al. (2014)'s random forests (Eslami et al., 2014)	-6.94	3.88

The MV kernel is defined in Section B.2. Here product (sum) of expected product kernels refers to a product (sum) of kernels, where each is an expected product kernel defined on one incoming message. Evidently, the Gaussian kernel on joint mean embeddings performs significantly better than other kernels. Besides the proposed method, we also compare the message prediction performance to Breiman's random forests (Breiman, 2001), extremely randomized trees (Geurts et al., 2006), and Eslami et al. (2014)'s random forests. We use scikit-learn toolbox for the extremely randomized trees and Breiman's random forests. For Eslami et al. (2014)'s random forests, we reimplemented the method as closely as possible according to the description given in Eslami et al. (2014). In all cases, the number of trees is set to 64. Empirically we observe that decreasing the log KL error below -8 will not yield a noticeable performance gain in the actual EP.

Figure 10: KL-divergence error versus predictive variance for predicting the mean of $m_{f \to z_i}$ (normal distribution) in the logistic factor problem.

To verify that the uncertainty estimates given by KJIT coincide with the actual predictive performance (i.e., accurate prediction when confident), we plot the predictive variance against the KL-divergence error on both the training and test sets. The results are shown in Fig. 10. The uncertainty estimates show a positive correlation with the KL-divergence errors. It is instructive to note that no point lies at the bottom right i.e., making a large error while being confident. The fact that the errors on the training set are roughly the same as the errors on the test set indicates that the operator does not overfit.

References

[sup1] L. Breiman. Random Forests. *Mach. Learn.*, 45(1):5–32, 2001.

[sup7] S. M. A. Eslami, D. Tarlow, P. Kohli, and J. Winn. Just-In-Time Learning for Fast and Flexible Inference. In *NIPS*, pages 154–162, 2014.

[sup3] P. Geurts, D. Ernst, and L. Wehenkel. Extremely Randomized Trees. *Mach. Learn.*, 63(1):3–42, 2006.

[sup17] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007.

[sup19] W. Rudin. *Fourier Analysis on Groups: Interscience Tracts in Pure and Applied Mathematics, No. 12*. Literary Licensing, LLC, 2013.

[sup21] A. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *ALT*, pages 13–31, 2007.