Generative Models and Model Criticism via Optimized Maximum Mean Discrepancy

Dougal J. Sutherland[∗ †] **Hsiao-Yu Tung**[†] **Heiko Strathmann**[∗] **Soumyajit De**[∗]
Aaditya Ramdas[‡] **Alex Smola**[†] **Arthur Gretton**[∗]
[∗] Gatsby Computational Neuroscience Unit, University College London
[†] School of Computer Science, Carnegie Mellon University
[‡] Departments of EECS and Statistics, University of California at Berkeley
dougal@gmail.com htung@cs.cmu.edu

Abstract

We propose a method to optimize the representation and distinguishability of samples from two probability distributions, by maximizing the estimated power of a statistical test based on the maximum mean discrepancy (MMD). This optimized MMD is applied to the setting of unsupervised learning by generative adversarial networks (GAN), in which a model attempts to generate realistic samples, and a discriminator attempts to tell these apart from data samples. In this context, the MMD may be used in two roles: first, as a discriminator, either directly on the samples, or on features of the samples. Second, the MMD can be used to evaluate the performance of a generative model, by testing the model's samples against a reference data set. In the latter role, the optimized MMD is particularly helpful, as it gives an interpretable indication of how the model and data distributions differ, even in cases where individual model samples are not easily distinguished either by eye or by classifier.

1 Introduction

Many problems in testing and learning require evaluating distribution similarity in high dimensions, and on structured data such as images or audio. When a complex generative model is learned, it is necessary to provide feedback on the quality of the samples produced. The generative adversarial network (Goodfellow et al., 2014; Gutmann et al., 2014) is a popular method for training generative models, where a rival discriminator attempts to distinguish model samples from reference data. Training of the generator and discriminator is interleaved, such that a saddle point is eventually reached in the joint loss.

A useful insight into the behavior of GANs is to note that when the discriminator is properly trained, the generator is tasked with minimizing the Jensen-Shannon divergence measure between the model and data distributions. When the model is insufficiently powerful to perfectly simulate the test data, as in most nontrivial settings, the choice of divergence measure is especially crucial: it determines which compromises will be made. A range of adversarial divergences were proposed by Huszar (2015), using a weight to interpolate between KL, inverse KL, and Jensen-Shannon. This weight may be interpreted as a prior probability of observing samples from the model or the real world: when there is a greater probability of model samples, we approach reverse KL and the model seeks out modes of the data distribution. When there is a greater probability of drawing from the data distribution, the model approaches the KL divergence, and tries to cover the full support of the data, at the expense of producing some samples in low probability regions.

This insight was further developed by Nowozin et al. (2016), who showed that a much broader range of f-divergences can be learned for the discriminator in adversarial models, based on the variational formulation of f-divergences of Nguyen et al. (2008). For a given f-divergence, the model learns the composition of the density ratio (of data to model density) with the derivative of f, by comparing generator and data samples. This provides a lower bound on the "true" divergence that would be obtained if the density ratio were perfectly known. In the event that the model is in a smaller class than the true data distribution, this broader family of divergences implements a variety of different approximations: some focus on individual modes of the true sample density, others try to cover the support. It is straightforward to visualize these properties in one or two dimensions (Nowozin et al., 2016, Figure 5), but in higher dimensions it becomes difficult to anticipate or visualize the behavior of these various divergences.

An alternative family of divergences are the integral probability metrics (Müller, 1997), which find a witness function to distinguish samples from P and Q.[1] A popular such class of witness functions in GANs is the maximum mean discrepancy (Gretton et al., 2012a), simultaneously proposed by Dziugaite et al. (2015) and Li et al. (2015). The architecture used in these two approaches is actually quite different: Dziugaite et al. use the MMD as a discriminator directly at the level of the generated and test images, whereas Li et al. apply the MMD on input features learned from an autoencoder, and share the decoding layers of the autoencoder with the generator network (see their Figure 1(b)). The generated samples have better visual quality in the latter method, but it becomes difficult to analyze and interpret the algorithm given the interplay between the generator and discriminator networks. In a related approach, Salimans et al. (2016) propose to use feature matching, where the generator is tasked with minimizing the squared distance between expected discriminator features under the model and data distributions, thus retaining the adversarial setting.

In light of these varied approaches to discriminator training, it is important to be able to evaluate quality of samples from a generator against reference data. An approach used in several studies is to obtain a Parzen window estimate of the density and compute the log-likelhiood (Goodfellow et al., 2014; Nowozin et al., 2016; Breuleux et al., 2011). Unfortunately, density estimates in such high dimensions are known to be very unreliable both in theory (Wasserman, 2006, Ch. 6) and in practice (Theis et al., 2016). We can instead ask humans to evaluate the generated images (Denton et al., 2015; Salimans et al., 2016), but while evaluators should be able to distinguish cases where the samples are over-dispersed (support of the model is too large), it may be more difficult to find under-dispersed samples (too concentrated at the modes), or imbalances in the proportions of different shapes, since the samples themselves will be plausible images. Recall that different divergence measures result in different degrees of mode-seeking: if we rely on human evaluation, we may tend towards always using divergences with under-dispersed samples.

We propose to use the MMD to distinguish generator and reference data, with features and kernels chosen to maximize the test power of the quadratic-time MMD of Gretton et al. (2012a). Optimizing MMD test power requires a sophisticated treatment due to the different form of the null and alternative distributions (Section 2). We also develop an efficient approach to obtaining quantiles of the MMD distribution under the null (Section 3). We demonstrate on simple artificial data that simply maximizing the MMD (as in Sriperumbudur et al., 2009) provides a less powerful test than our approach of explicitly maximizing test power. Our procedure applies even when our definition of the MMD is computed on features of the inputs, since these can also be trained by power maximization.

When designing an optimized MMD test, we should choose a kernel family that allows us to visualize where the probability mass of the two samples differs most. In our experiments on GAN performance evaluation, we use an automatic relevance determination (ARD) kernel over the output dimensions, and learn which coordinates differ meaningfully by finding which kernels retain significant bandwidth when the test power is optimized. We may further apply the method of Lloyd & Ghahramani (2015, Section 5) to visualize the witness function associated with this MMD, by finding those model and data samples occurring at the maxima and minima of the witness function (i.e., the samples from one distribution least likely to be in high probability regions of the other). The optimized witness function gives a test with greater power than a standard RBF kernel, suggesting that the associated witness function peaks are an improved representation of where the distributions differ. We also propose a novel generative model based on the feature matching idea of Salimans et al. (2016), using MMD rather than their “minibatch discrimination” heuristic, for a more principled and more stable enforcement of sample diversity, without requiring labeled data.

2 Maximizing Test Power of a Quadratic MMD Test

Our methods rely on optimizing the power of a two-sample test over the choice of kernel. We first describe how to do this, then review alternative kernel selection approaches.

[1] Only the total variation distance is both an f-divergence and an IPM (Sriperumbudur et al., 2012).

2.1 MMD AND TEST POWER

We will begin by reviewing the maximum mean discrepancy and its use in two-sample tests. Let k be the kernel of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ of functions on a set $\mathcal{X}$. We assume that k is measurable and bounded, $\sup_{x \in \mathcal{X}} k(x,x) < \infty$. Then the MMD in $\mathcal{H}_k$ between two distributions P and Q over $\mathcal{X}$ is (Gretton et al., 2012a):

$$\text{MMD}_k^2(P,Q) := \mathbb{E}_{x,x'}\left[k(x,x')\right] + \mathbb{E}_{y,y'}\left[k(y,y')\right] - 2\,\mathbb{E}_{x,y}\left[k(x,y)\right] \tag{1}$$

where $x, x' \overset{iid}{\sim} P$ and $y, y' \overset{iid}{\sim} Q$. Many kernels, including the popular Gaussian RBF, are *characteristic* (Fukumizu et al., 2008; Sriperumbudur et al., 2010), which implies that the MMD is a metric, and in particular that $\text{MMD}_k(P,Q) = 0$ if and only if $P = Q$, so that tests with any characteristic kernel are consistent. That said, different characteristic kernels will yield different test powers for finite sample sizes, and so we wish to choose a kernel k to maximize the test power. Below, we will usually suppress explicit dependence on k.

Given $X = \{X_1, \dots, X_m\} \overset{iid}{\sim} P$ and $Y = \{Y_1, \dots, Y_m\} \overset{iid}{\sim} Q$,[2] one estimator of $\text{MMD}(P,Q)$ is

$$\widehat{\text{MMD}}_{\text{U}}^2(X,Y) := \frac{1}{\binom{m}{2}} \sum_{i \neq i'} k(X_i, X_{i'}) + \frac{1}{\binom{m}{2}} \sum_{j \neq j'} k(Y_j, Y_{j'}) - \frac{2}{\binom{m}{2}} \sum_{i \neq j} k(X_i, Y_j). \tag{2}$$

This estimator is unbiased, and has nearly minimal variance among unbiased estimators (Gretton et al., 2012a, Lemma 6).

Following Gretton et al. (2012a), we will conduct a hypothesis test with null hypothesis $H_0 : P = Q$ and alternative $H_1 : P \neq Q$, using test statistic $m\,\widehat{\text{MMD}}_{\text{U}}^2(X,Y)$. For a given allowable probability of false rejection α, we choose a test threshold c_α and reject H_0 if $m\,\widehat{\text{MMD}}_{\text{U}}^2(X,Y) > c_\alpha$.

Under $H_0 : P = Q$, $m\,\widehat{\text{MMD}}_{\text{U}}^2(X,Y)$ converges asymptotically to a distribution that depends on the unknown distribution P (Gretton et al., 2012a, Theorem 12); we thus cannot evaluate the test threshold c_α in closed form. We instead estimate a data-dependent threshold $\hat{c}_\alpha$ via permutation: randomly partition the data $X \cup Y$ into X' and Y' many times, evaluate $m\,\widehat{\text{MMD}}_{\text{U}}^2(X',Y')$ on each split, and estimate the $(1-\alpha)$th quantile $\hat{c}_\alpha$ from these samples. Section 3 discusses efficient computation of this process.

We now describe a mechanism to choose the kernel k so as to maximize the power of its associated test. First, note that under the alternative $H_1 : P \neq Q$, $\widehat{\text{MMD}}_{\text{U}}^2$ is asymptotically normal,

$$\frac{\widehat{\text{MMD}}_{\text{U}}^2(X,Y) - \text{MMD}^2(P,Q)}{\sqrt{V_m(P,Q)}} \xrightarrow{D} \mathcal{N}(0,1), \tag{3}$$

where $V_m(P,Q)$ denotes the asymptotic variance of the $\widehat{\text{MMD}}_{\text{U}}^2$ estimator for samples of size m from P and Q (Serfling, 1980). The power of our test is thus, using $\Pr_1$ to denote probability under H_1,

$$\begin{aligned} \Pr_1\left(m\,\widehat{\text{MMD}}_{\text{U}}^2(X,Y) > \hat{c}_\alpha\right) &= \Pr_1\left(\frac{\widehat{\text{MMD}}_{\text{U}}^2(X,Y) - \text{MMD}^2(P,Q)}{\sqrt{V_m(P,Q)}} > \frac{\hat{c}_\alpha/m - \text{MMD}^2(P,Q)}{\sqrt{V_m(P,Q)}}\right) \\ &\to \Phi\left(\frac{\text{MMD}^2(P,Q)}{\sqrt{V_m(P,Q)}} - \frac{c_\alpha}{m\sqrt{V_m(P,Q)}}\right) \end{aligned} \tag{4}$$

where Φ is the CDF of the standard normal distribution. The second step follows by (3) and the convergence of $\hat{c}_\alpha \to c_\alpha$ (Alba Fernández et al., 2008). Test power is therefore maximized by maximizing the argument of Φ: i.e. increasing the ratio of $\text{MMD}^2(P,Q)$ to $\sqrt{V_m(P,Q)}$, and reducing the ratio of c_α to $m\sqrt{V_m(P,Q)}$.

For a given kernel k, V_m is $O(m^{-1})$, while both c_α and MMD^2 are constants. Thus the first term is $O(\sqrt{m})$, and the second is $O(1/\sqrt{m})$. Two situations therefore arise: when m is small relative to the difference in P and Q (i.e., we are close to the null), both terms need to be taken into acccount

[2] We assume for simplicity that the number of samples from the two distributions is equal.

to maximize test power. Here, we propose to maximize (4) using the efficient computation of $\hat{c}_\alpha$ in Section 3. As m grows, however, we can asymptotically maximize the power of the test by choosing a kernel k that maximizes the t-statistic $t_k(P, Q) := \text{MMD}_k^2(P, Q) / \sqrt{V_m^{(k)}(P, Q)}$. In practice, we maximize an estimator of $t_k(P, Q)$ given by $\hat{t}_k(X, Y) := \widehat{\text{MMD}}_\text{U}^2(X, Y) / \sqrt{\widehat{V}_m(X, Y)}$, with $\widehat{V}_m(X, Y)$ discussed shortly.

To maintain the validity of the hypothesis test, we will need to divide the observed data X and Y into a "training sample," used to choose the kernel, and a "testing sample," used to perform the final hypothesis test with the learned kernel.

We next consider families of kernels over which to optimize. The most common kernels used for MMD tests are standard kernels from the literature, e.g. the Gaussian RBF, Matérn, or Laplacian kernels. It is the case, however, that for any function $z : \mathcal{X}_1 \to \mathcal{X}_2$ and any kernel $\kappa : \mathcal{X}_2 \times \mathcal{X}_2 \to \mathbb{R}$, the composition $\kappa \circ z$ is also a kernel on $\mathcal{X}_1$.[3] We can thus choose a function z to extract meaningful features of the inputs, and use a standard kernel κ to compare those features. We can select such a function z (as well as κ) by performing kernel selection on the family of kernels $\kappa \circ z$. To do so, we merely need to maximize $\hat{t}_{\kappa \circ z}(X, Y)$ through standard optimization techniques based on the gradient of $\hat{t}_{\kappa \circ z}$ with respect to the parameterizations of z and κ.

We now give an expression for an empirical estimate $\widehat{V}_m$ of the variance $V_m(P, Q)$ that appears in our test power. This estimate is similar to that given by Bounliphone et al. (2016, Appendix A.1), but incorporates second-order terms and corrects some small sources of bias. Though the expression is somewhat unwieldy, it is defined by various sums of the kernel matrices and is differentiable with respect to the kernel k.

$V_m(P, Q)$ is given in terms of expectations of k under P and Q in Appendix A. We replace these expectations with finite-sample averages, giving us the required estimator. Define matrices K_{XY}, $\tilde{K}_{XX}$, and $\tilde{K}_{YY}$ by $(K_{XY})_{i,j} = k(X_i, Y_j)$, $(\tilde{K}_{XX})_{ii} = 0$, $(\tilde{K}_{XX})_{ij} = k(X_i, X_j)$ for $i \neq j$, and $\tilde{K}_{YY}$ similarly to $\tilde{K}_{XX}$. Let e be a vector of ones. Then an unbiased estimator for $V_m(P, Q)$ is:

$$
\begin{aligned}
\widehat{V}_m := {} & \frac{2}{m^2(m-1)^2}\left(2\|\tilde{K}_{XX}e\|^2 - \|\tilde{K}_{XX}\|_F^2 + 2\|\tilde{K}_{YY}e\|^2 - \|\tilde{K}_{YY}\|_F^2\right) \\
& - \frac{4m-6}{m^3(m-1)^3}\left[\left(e^\mathsf{T}\tilde{K}_{XX}e\right)^2 + \left(e^\mathsf{T}\tilde{K}_{YY}e\right)^2\right] + \frac{4(m-2)}{m^3(m-1)^2}\left(\|K_{XY}e\|^2 + \|K_{XY}^\mathsf{T}e\|^2\right) \\
& - \frac{4(m-3)}{m^3(m-1)^2}\|K_{XY}\|_F^2 - \frac{8m-12}{m^5(m-1)}\left(e^\mathsf{T}K_{XY}e\right)^2 \qquad (5) \\
& + \frac{8}{m^3(m-1)}\left(\frac{1}{m}\left(e^\mathsf{T}\tilde{K}_{XX}e + e^\mathsf{T}\tilde{K}_{YY}e\right)\left(e^\mathsf{T}K_{XY}e\right) - e^\mathsf{T}\tilde{K}_{XX}K_{XY}e - e^\mathsf{T}\tilde{K}_{YY}K_{XY}^\mathsf{T}e\right).
\end{aligned}
$$

2.2 Other Approaches to MMD Kernel Selection

The most common practice in performing two-sample tests with MMD is to use a Gaussian RBF kernel, with bandwidth set to the median pairwise distance among the joint data. This heuristic often works well, but fails when the scale on which P and Q vary differs from the scale of their overall variation (as in the synthetic experiment of Section 4). Ramdas et al. (2015a;b) study the power of the median heuristic in high-dimensional problems, and justify its use for the case where the means of P and Q differ.

An early heuristic for improving test power was to simply maximize $\widehat{\text{MMD}}_\text{U}^2$. Sriperumbudur et al. (2009) proved that, for certain classes of kernels, this yields a consistent test. As further shown by Sriperumbudur et al., however, maximizing MMD amounts to minimizing training *classification error* under linear loss. Comparing with (4), this is plainly not an optimal approach for maximizing *test* power, since variance is ignored.[4] One can also consider maximizing criteria based on cross validation

[3] If z is injective and κ characteristic, then $\kappa \circ z$ is characteristic. Whether any fixed $\kappa \circ z$ is consistent, however, is less relevant than the power of the $\kappa \circ z$ we choose — which is what we maximize.

[4] With regards to classification vs testing: there has been initial work by Ramdas et al. (2016), who study the simplest setting of the two multivariate Gaussians with known covariance matrices. Here, one can use linear

(Sugiyama et al., 2011; Gretton et al., 2012b; Strathmann, 2012). This approach is not differentiable, and thus difficult to maximize among more than a fixed set of candidate kernels. Moreover, where this cross-validation is used to maximize the MMD on a validation set (as in Sugiyama et al., 2011), it again amounts to maximizing classification performance rather than test performance, and is suboptimal in the latter setting (Gretton et al., 2012b, Figure 1). Finally, Gretton et al. (2012b) previously studied direct optimization of the power of an MMD test for a streaming estimator of the MMD, for which optimizing the ratio of the empirical statistic to its variance also optimizes test power. This streaming estimator uses data very inefficiently, however, often requiring m^2 samples to achieve power comparable to tests based on $\widehat{\text{MMD}}_U^2$ with m samples (Ramdas et al., 2015a).

3 Efficient Implementation of Permutation Tests for $\widehat{\text{MMD}}_U^2$

Practical implementations of tests based on $\widehat{\text{MMD}}_U^2$ require efficient estimates of the test threshold $\hat{c}_\alpha$. There are two known test threshold estimates that lead to a consistent test: the permutation test mentioned above, and a more sophisticated null distribution estimate based on approximating the eigenspectrum of the kernel, previously reported to be faster than the permutation test (Gretton et al., 2009). In fact, the relatively slow reported performance of the permutation approach was due to the naive Matlab implementation of the permutation test in the code accompanying Gretton et al. (2012a), which creates a new copy of the kernel matrix for every permutation. We show here that, by careful design, permutation thresholds can be computed substantially faster – even when compared to parallelized state-of-the-art spectral solvers (not used by Gretton et al.).

First, we observe that we can avoid copying the kernel matrix simply by generating permutation indices for each null sample and accessing the precomputed kernel matrix in permuted order. In practice, however, this does not give much performance gain due to the random nature of memory-access which conflicts with how modern CPUs implement caching. Second, if we rather maintain an inverse map of the permutation indices, we can easily traverse the matrix in a sequential fashion. This approach exploits the hardware prefetchers and reduces the number of CPU cache misses from almost 100% to less than 10%. Furthermore, the sequential access pattern of the kernel matrix enables us to invoke multiple threads for computing the null samples, each traversing the matrix sequentially, without compromising the locality of reference in the CPU cache.

We consider an example problem of computing the test using 200 null distribution samples on $m = 2000$ two-dimensional samples, comparing a Gaussian to a Laplace distribution with matched moments. We compare our optimized permutation test against a spectral test using the highly-optimized (and proprietary) state-of-the-art spectral solver of Intel's MKL library (Intel, 2003–17). All results are averaged over 30 runs; the variance across runs was negligible.

Figure 1 (left) shows the obtained speedups as the number of computing threads grow for $m = 2000$. Our implementation is not only faster on a single thread, but also saturates more slowly as the number of threads increases. Figure 1 (right) shows timings for increasing problem sizes (i.e. m) when using all available system threads (here 24). For larger problems, our permutation implementation (scaling as $\mathcal{O}(m^2)$) is an order of magnitude faster than the spectral test (scaling as $\mathcal{O}(m^3)$). For smaller problems (for which Gretton et al. suggested the spectral test), there is still a significant performance increase.

For further reference, we also report timings of available non-parallelized implementations for $m = 2000$, compared to our version's 12s in Figure 1 (left): 87s for an open-sourced spectral test in Shogun using eigen3 (Sonnenburg et al., 2016; Guennebaud et al., 2010), 381s for the reference Matlab spectral implementation (Gretton et al., 2012a), and 182s for a naive Python permutation test that partly avoids copying via masking. (All of these times also exclude kernel computation.)

classifiers, and the two sample test boils down to testing for differences in means. In this setting, when the classifier is chosen to be Fisher's LDA, then using the classifier accuracy on held-out data as a test statistic turns out to be minimax optimal in "rate" (dependence on dimensionality and sample size) but not in constants, meaning that there do exist tests which achieve the same power with fewer samples. The result has been proved only for this statistic and setting, however, and generalization to other statistics and settings is an open question.

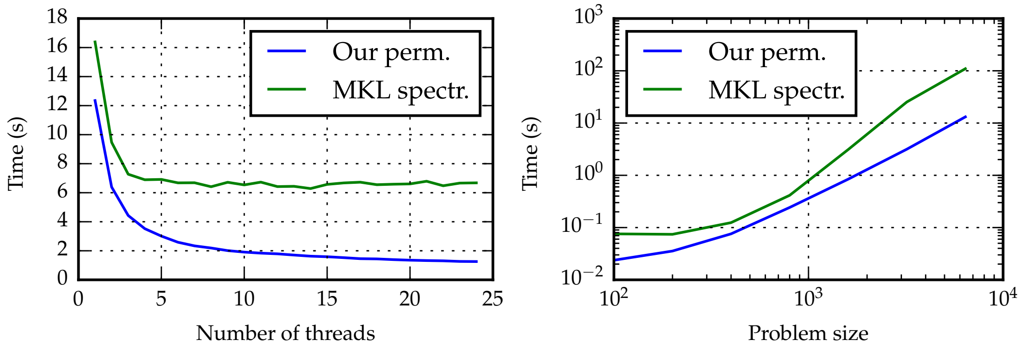


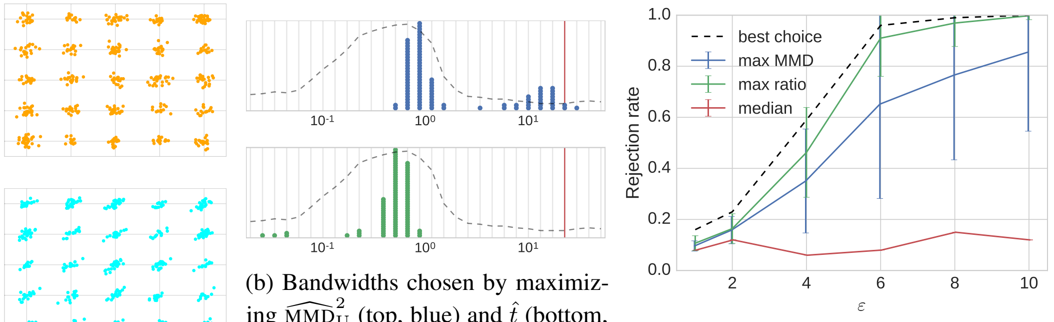
Figure 1: Runtime comparison for sampling the null distribution. We compare our optimized permutation approach to the spectral method using Intel's MKL spectral solver. Time spent precomputing the kernel matrix is not included. *Left:* Increasing number of threads for fixed problem size $m = 2000$. Single-threaded times of other implementations: Matlab reference spectral 381s, Python permutation 182s, Shogun spectral (eigen3) 87s. *Right:* Increasing problem sizes using the maximum number of 24 system threads.

4 Experiments

Code for these experiments is available at `github.com/dougalsutherland/opt-mmd`.

Synthetic data We consider the problem of bandwidth selection for Gaussian RBF kernels on the Blobs dataset of Gretton et al. (2012b). P here is a 5×5 grid of two-dimensional standard normals, with spacing 10 between the centers. Q is laid out identically, but with covariance $\frac{\varepsilon-1}{\varepsilon+1}$ between the coordinates (so that the ratio of eigenvalues in the variance is ε.) Figure 2a shows two samples from X and Y with $\varepsilon = 6$. Note that when $\varepsilon = 1$, $P = Q$.

For $\varepsilon \in \{1, 2, 4, 6, 8, 10\}$, we take $m = 500$ samples from each distribution and compute $\widehat{\mathrm{MMD}}^2_{\mathrm{U}}(X, Y)$, $\widehat{V}_m(X, Y)$, and $\hat{c}_{0.1}$ using 1 000 permutations, for Gaussian RBF kernels with each of 30 bandwidths. We repeat this process 100 times. Figure 2b shows that the median heuristic always chooses too large a bandwidth. When maximizing MMD alone, we see a bimodal distribution of bandwidths, with a significant number of samples falling into the region with low test power. The variance of $\widehat{\mathrm{MMD}}^2_{\mathrm{U}}$ is much higher in this region, however, hence optimizing the ratio $\hat{t}$ never returns these bandwidths. Figure 2c shows that maximizing $\hat{t}$ outperforms maximizing the MMD across a variety of problem parameters, and performs near-optimally.



(a) Samples of size 500 with $\varepsilon = 6$ from P (top) and Q (bottom).

(b) Bandwidths chosen by maximizing $\widehat{\mathrm{MMD}}^2_{\mathrm{U}}$ (top, blue) and $\hat{t}$ (bottom, green), as well as the median heuristic (red), for $\varepsilon = 6$. Gray lines show the power of each bandwidth: $\sigma = 0.67$ had power 96%, $\sigma = 10$ had 10%.

(c) Mean and standard deviation of rejection rate as ε increases. "Best choice" shows the mean power of the bandwidth with highest rejection rate for each problem.

Figure 2: Results for the Blobs problem. Maximizing $\hat{t}$ performs near-optimally.

Model criticism As an example of a real-world two-sample testing problem, we will consider distinguishing the output of a generative model from the reference distribution it attempts to reproduce. We will use the semi-supervised GAN model of Salimans et al. (2016), trained on the MNIST dataset of handwritten images.[5] True samples from the dataset are shown in Figure 3a; samples from the learned model are in Figure 3b. Salimans et al. (2016) called their results "completely indistinguishable from dataset images," and reported that annotators on Mechanical Turk were able to distinguish samples only in 52.4% of cases. Comparing the results, however, there are several pixel-level artifacts that make distinguishing the datasets trivial; our methods can pick up on these quickly.

To make the problem more interesting, we discretized the sampled pixels into black or white (which barely changes the images visually). The samples are then in $\{0, 1\}^{28\times28}$. We trained an automatic relevance determination (ARD)-type kernel: in the notation of Section 2.1, z scales each pixel by some learned value, and k is a Gaussian RBF kernel with a learned global bandwidth. We optimized $\hat{t}$ on 2 000 samples in batches of size 500 using the Adam optimizer (Kingma & Ba, 2015), where the learned weights are visualized in Figure 3c. This network has essentially perfect discriminative power: testing it on 100 different samples with 1000 permutations for each test, in 98 cases we obtained p-values of 0.000 and twice got 0.001. By contrast, using an RBF kernel with a bandwidth optimized by maximizing the t statistic gave a less powerful test: the worst p-value in 100 repetitions was 0.135, with power 57% at the $\alpha = 0.01$ threshold. An RBF kernel based on the median heuristic, which here found a bandwidth five times the size of the t-statistic-optimized bandwidth, performed worse still: three out of 100 repetitions found a p-value of exactly 1.000, and power at the .01 threshold was 42%. The learned weights show that the model differs from the true dataset along the outsides of images, as well as along a vertical line in the center.

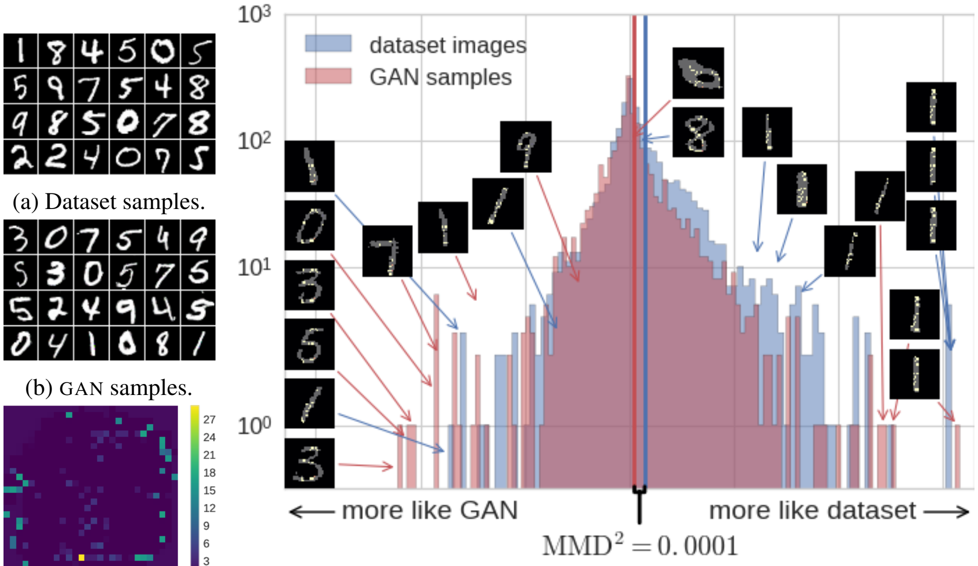
We can investigate these results in further detail using the approach of Lloyd & Ghahramani (2015), considering the witness function associated with the MMD, which has largest amplitude where the probability mass of the two samples is most different. Thus, samples falling at maxima and minima of the witness function best represent the difference in the distributions. The value of the witness function on each sample is plotted in Figure 3d, along with some images with different values of the witness function. Apparently, the GAN is slightly overproducing images resembling the /-like digits on the left, while underproducing vertical 1s. It is not the case that the GAN is simply underproducing 1s in general: the p-values of a χ^2 contingency test between the outputs of digit classifiers on the two distributions are uniform. This subtle difference in proportions among types of digits would be quite difficult for human observers to detect. Our testing framework allows the model developer to find such differences and decide whether to act on them. One could use a more complex representation function z to detect even more subtle differences between distributions.

GAN criterion We now demonstrate the use of MMD as a training criterion in GANs. We consider two basic approaches, and train on MNIST.[6] First, the generative moment matching network (GMMN; Figure 4a) approach (Li et al., 2015; Dziugaite et al., 2015) uses an MMD statistic computed with an RBF kernel directly on the images as the discriminator of a GAN model. The t-GMMN (Figure 4b) has the generator minimize the $\hat{t}_k$ statistic for a fixed kernel.[7] Compared to standard GMMNs, the t-GMMN more directly attempts to make the distributions *indistinguishable* under the kernel function; it avoids a situation like that of Figure 3d, where although the MMD value is quite small, the two distributions are perfectly distinguishable due to the small variance.

[5] We used their code for a minibatch discrimination GAN, with 1000 labels, and chose the best of several runs.

[6] **Implementation details:** We used the architecture of Li et al. (2015): the generator consists of fully connected layers with sizes $10, 64, 256, 256, 1024, 784$, each with ReLU activations except the last, which uses sigmoids. The kernel function for GMMNs is a sum of Gaussian RBF kernels with fixed bandwidths $2, 5, 10, 20, 40, 80$. For the feature matching GAN, we use a discriminator with fully connected layers of size $512, 256, 256, 128, 64$, each with sigmoid activation. We then concatenate the raw image and each layer of features as input to the same mixture of RBF kernels as for GMMNs. We optimize with SGD. Initialization for all parameters are Gaussian with standard deviation 0.1 for the GMMNs and 0.2 for feature matching. Learning rates are $2, 0.02, 0.5$, respectively. Learning rate for the feature matching discriminator is set to 0.01. All experiments are run for 50 000 iterations and use a momentum optimizer with with momentum 0.9.

[7] One could additionally update the kernel adversarially, by maximizing the $\hat{t}_k$ statistic based on generator samples, but we had difficulty in optimizing this model: the interplay between generator and discriminator adds some difficulty to this task.



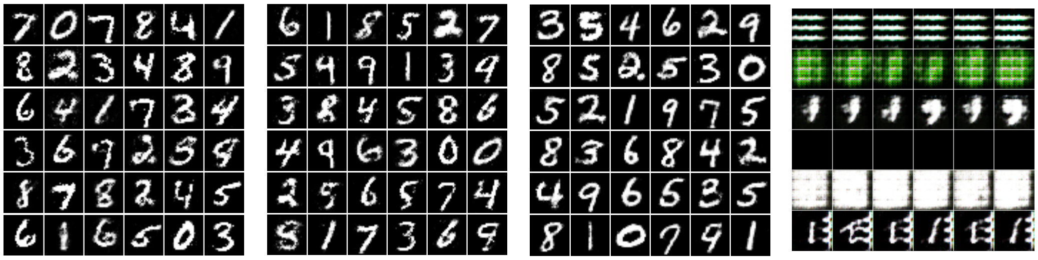
(a) Dataset samples.

(b) GAN samples.

(c) ARD weights.

(d) The witness function $\mathbb{E}_{x \sim P}\, k(x, \cdot) - \mathbb{E}_{y \sim Q}\, k(y, \cdot)$ evaluated on a test set. Images are shown with ARD weights highlighted. Vertical lines show distribution means; the distance between them is small but highly consistent.

Figure 3: Model criticism of Salimans et al. (2016)'s semi-supervised GAN on MNIST.



(a) GMMN

(b) t-GMMN

(c) Feature matching

(d) Improved GAN

Figure 4: MNIST digits from various models. Part d shows six runs of the minibatch discrimination model of Salimans et al. (2016), trained without labels — the same model that, with labels, generated Figure 3b. (The third row is the closest we got the model to generating digits without any labels.)

Next, feature matching GANs (Figure 4c) train the discriminator as a classifier like a normal GAN, but train the generator to minimize the MMD between generator samples and reference samples with a kernel computed on intermediate features of the discriminator. Salimans et al. (2016) proposed feature matching using the mean features at the top of the discriminator (effectively using an MMD with a linear kernel); we instead use MMD with a mixture of RBF kernels, ensuring that the full feature distributions match, rather than just their means. This helps avoid the common failure mode of GANs where the generator collapses to outputting a small number of samples considered highly realistic by the discriminator. Using the MMD-based approach, however, no single point can approximate the feature *distribution*. The minibatch discrimination approach of Salimans et al. (2016) attempts to solve the same problem, by introducing features measuring the similarity of each sample to a selection of other samples, but we were unable to get it to work without labels to force the discriminator in a reasonable direction; Figure 4d demonstrates some of those failures, with each row showing six samples from each of six representative runs of the model.

Acknowledgements

We would like to thank Tim Salimans, Ian Goodfellow, and Wojciech Zaremba for providing their code and for gracious assistance in using it, as well as Jeff Schneider for helpful discussions.

References

V. Alba Fernández, M. Jiménez-Gamero, and J. Muñoz Garcia. A test for the two-sample problem based on empirical characteristic functions. *Computational Statistics & Data Analysis*, 52:3730–3748, 2008.

Wacha Bounliphone, Eugene Belilovsky, Matthew B. Blaschko, Ioannis Antonoglou, and Arthur Gretton. A test of relative similarity for model selection in generative models. In *International Conference on Learning Representations*, 2016. arXiv:1511.04581.

Olivier Breuleux, Yoshua Bengio, and Pascal Vincent. Quickly generating representative samples from an rbm-derived process. *Neural Computation*, 23(8):2053–2073, 2011.

Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, pp. 1486–1494. 2015.

Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Uncertainty in Artificial Intelligence*, 2015. arXiv:1505.03906.

Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems*, 2008.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. arXiv:1406.2661.

Arthur Gretton, Kenji Fukumizu, Zaid Harchaoui, and Bharath K. Sriperumbudur. A fast, consistent kernel two-sample test. In *Advances in Neural Information Processing Systems*, pp. 673–681, 2009.

Arthur Gretton, Karsten M. Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alex J. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13, 2012a.

Arthur Gretton, Bharath Sriperumbudur, Dino Sejdinovic, Heiko Strathmann, and Massimiliano Pontil. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems*, 2012b.

Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

Michael U. Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Statistical inference of intractable generative models via classification. 2014. arXiv:1407.4981.

Ferenc Huszar. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? 2015. arXiv:1511.05101.

MKL Intel. Intel math kernel library, 2003–17.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. arXiv:1412.6980.

Yujia Li, Kevin Swersky, and Richard Zemel. Generative moment matching networks. In *Uncertainty in Artificial Intelligence*, 2015. arXiv:1502.02761.

James Robert Lloyd and Zoubin Ghahramani. Statistical model criticism using kernel two sample tests. In *Advances in Neural Information Processing Systems*, pp. 829–837, 2015.

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. On optimal quantization rules in some problems in sequential decentralized detection. *IEEE Transactions on Information Theory*, 54(7):3285–3295, 2008.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, 2016.

Aaaditya Ramdas, Sashank Reddi, Barnabás Póczos, Aarti Singh, and Larry Wasserman. Adaptivity and Computation-Statistics Tradeoffs for Kernel and Distance based High Dimensional Two Sample Testing, 2015a. arXiv:1508.00655.

Aaditya Ramdas, Sashank J. Reddi, Barnabás Póczos, Aarti Singh, and Larry Wasserman. On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. In *AAAI Conference on Artificial Intelligence*, 2015b. arXiv:1406.2083.

Aaditya Ramdas, Aarti Singh, and Larry Wasserman. Classification accuracy as a proxy for two sample testing, 2016. arXiv:1602.02210.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 2016. arXiv:1606.03498.

Robert J. Serfling. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, 1980.

Soeren Sonnenburg, Heiko Strathmann, Sergey Lisitsyn, Viktor Gal, Fernando J. Iglesias García, Wu Lin, Chiyuan Zhang, Soumyajit De, frx, tklein23, Evgeniy Andreev, JonasBehr, sploving, Parijat Mazumdar, Christian Widmer, Abhijeet Kislay, Kevin Hughes, Roman Votyakov, khalednasr, Saurabh Mahindre, Alesis Novik, Abinash Panda, Evangelos Anagnostopoulos, Liang Pang, serialhex, Alex Binder, Sanuj Sharma, Michal Uřičář, Björn Esser, and Daniel Pyrathon. shogun: Shogun 4.1.0 - tajinohi no agatamori, May 2016.

Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Gert R. G. Lanckriet, and Bernhard Schölkopf. Kernel choice and classifiability for RKHS embeddings of probability distributions. In *Advances in Neural Information Processing Systems*, 2009.

Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Gert R. G. Lanckriet, and Bernhard Schölkopf. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010.

Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

Heiko Strathmann. *Adaptive Large-Scale Kernel Two-Sample Testing*. M.Sc. thesis, University College London, 2012.

Masashi Sugiyama, Taiji Suzuki, Yuta Itoh, Takafumi Kanamori, and Manabu Kimura. Least-squares two-sample test. *Neural Networks*, 24(7):735–751, sep 2011.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016. arXiv:1511.01844.

Larry Wasserman. *All of Nonparametric Statistics*. Springer, 2006.

A Variance of the Pairwise MMD Estimator

We will now derive the variance V_m of $\widehat{\mathrm{MMD}}^2_{\mathrm{U}}(X, Y)$. This derivation is similar to Appendix A of Bounliphone et al. (2016), except that we do not drop the second-order term, since it gives little computational advantage to do so. We further remove several small biases in their estimator: these have no effect for moderate to large sample sizes, but can have a discernible effect when the sample size is small.

Recall that our kernel k corresponds to a reproducing kernel Hilbert space $\mathcal{H}$ such that there exists a function φ with $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}$. Define $\mu_P := \mathbb{E}_{x\sim P}[\varphi(x)]$, $\mu_Q := \mathbb{E}_{y\sim Q}[\varphi(y)]$ to be the kernel mean embeddings of distributions P and Q; we then have that $\mathrm{MMD}(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}}$. Below, we will supress the explicit dependence on $\mathcal{H}$ for brevity.

Let $v := (x, y)$ and

$$h(v_1, v_2) := k(x_1, x_2) + k(y_1, y_2) - k(x_1, y_2) - k(x_2, y_1).$$

Then

$$\widehat{\mathrm{MMD}}^2_{\mathrm{U}}(X, Y) = \frac{1}{\binom{m}{2}} \sum_{i\neq j} h(v_i, v_j),$$

and by the theory of U-statistics (Serfling, 1980, Chapter 5) we have that

$$\mathrm{Var}\left[\widehat{\mathrm{MMD}}^2_{\mathrm{U}}(X, Y)\right] \to \frac{4(m-2)}{m(m-1)}\zeta_1 + \frac{2}{m(m-1)}\zeta_2,$$

where

$$\zeta_1 := \mathrm{Var}_{v_1}\left[\mathbb{E}_{v_2}\left[h(v_1, v_2)\right]\right], \qquad \zeta_2 := \mathrm{Var}_{v_1,v_2}\left[h(v_1, v_2)\right].$$

The first-order term ζ_1 is:

$$\begin{aligned}
\zeta_1 &= \mathrm{Var}_{v_1}\left[\mathbb{E}_{v_2}\left[h(v_1, v_2)\right]\right] \\
&= \mathrm{Var}_{v}\left[\langle \varphi(x), \mu_P\rangle + \langle \varphi(y), \mu_Q\rangle - \langle \varphi(x), \mu_Q\rangle - \langle \mu_P, \varphi(y)\rangle\right] \\
&= \mathrm{Var}\left[\langle \varphi(x), \mu_P\rangle\right] + \mathrm{Var}\left[\langle \varphi(y), \mu_Q\rangle\right] + \mathrm{Var}\left[\langle \varphi(x), \mu_Q\rangle\right] + \mathrm{Var}\left[\langle \mu_P, \varphi(y)\rangle\right] \\
&\quad - 2\,\mathrm{Cov}\left(\langle \varphi(x), \mu_P\rangle, \langle \varphi(x), \mu_Q\rangle\right) - 2\,\mathrm{Cov}\left(\langle \varphi(y), \mu_Q\rangle, \langle \mu_P, \varphi(y)\rangle\right).
\end{aligned}$$

Note that

$$\begin{aligned}
\mathrm{Var}\left[\langle \varphi(a), \mu_B\rangle\right] &= \mathbb{E}\left[\langle \varphi(a), \mu_B\rangle^2\right] - \langle \mu_A, \mu_B\rangle^2 \\
\mathrm{Cov}\left(\langle \varphi(a), \mu_B\rangle, \langle \varphi(a), \mu_C\rangle\right) &= \mathbb{E}\left[\langle \varphi(a), \mu_B\rangle\langle \varphi(a), \mu_C\rangle\right] - \langle \mu_A, \mu_B\rangle\langle \mu_A, \mu_C\rangle
\end{aligned}$$

so that

$$\begin{aligned}
\zeta_1 = \;& \mathbb{E}\left[\langle \varphi(x), \mu_P\rangle^2\right] - \langle \mu_P, \mu_P\rangle^2 \\
&+ \mathbb{E}\left[\langle \varphi(y), \mu_Q\rangle^2\right] - \langle \mu_Q, \mu_Q\rangle^2 \\
&+ \mathbb{E}\left[\langle \varphi(x), \mu_Q\rangle^2\right] - \langle \mu_P, \mu_Q\rangle^2 \\
&+ \mathbb{E}\left[\langle \varphi(y), \mu_P\rangle^2\right] - \langle \mu_Q, \mu_P\rangle^2 \\
&- 2\,\mathbb{E}\left[\langle \varphi(x), \mu_P\rangle\langle \varphi(x), \mu_Q\rangle\right] + 2\langle \mu_P, \mu_P\rangle\langle \mu_P, \mu_Q\rangle \\
&- 2\,\mathbb{E}\left[\langle \varphi(y), \mu_Q\rangle\langle \varphi(y), \mu_P\rangle\right] + 2\langle \mu_Q, \mu_Q\rangle\langle \mu_P, \mu_Q\rangle,
\end{aligned}$$

agreeing with (19) of Bounliphone et al. (2016).

We can similarly compute the second-order term ζ_2 as:

$$\begin{aligned}
\zeta_2 &= \mathrm{Var}\left[h(v_1, v_2)\right] \\
&= \mathrm{Var}\left[k(x_1, x_2) + k(y_1, y_2) - k(x_1, y_2) - k(x_2, y_1)\right] \\
&= \mathrm{Var}\left[k(x_1, x_2)\right] + \mathrm{Var}\left[k(y_1, y_2)\right] + \mathrm{Var}\left[k(x_1, y_2)\right] + \mathrm{Var}\left[k(x_2, y_1)\right] \\
&\quad - 2\,\mathrm{Cov}\left(k(x_1, x_2), k(x_1, y_2)\right) - 2\,\mathrm{Cov}\left(k(x_1, x_2), k(x_2, y_1)\right) \\
&\quad - 2\,\mathrm{Cov}\left(k(y_1, y_2), k(x_1, y_2)\right) - 2\,\mathrm{Cov}\left(k(y_1, y_2), k(x_2, y_1)\right) \\
&= \mathrm{Var}\left[k(x_1, x_2)\right] + \mathrm{Var}\left[k(y_1, y_2)\right] + 2\,\mathrm{Var}\left[k(x, y)\right]
\end{aligned}$$

$$
\begin{aligned}
&- 4\operatorname{Cov}\left(k(x_1,x_2),k(x_1,y)\right) - 4\operatorname{Cov}\left(k(y_1,y_2),k(y_1,x)\right)\\
&= \mathbb{E}\left[k(x_1,x_2)^2\right] - \langle\mu_P,\mu_P\rangle^2\\
&\quad + \mathbb{E}\left[k(y_1,y_2)^2\right] - \langle\mu_Q,\mu_Q\rangle^2\\
&\quad + 2\,\mathbb{E}\left[k(x,y)^2\right] - 2\langle\mu_P,\mu_Q\rangle^2\\
&\quad - 4\,\mathbb{E}\left[\langle\varphi(x),\mu_P\rangle\langle\varphi(x),\mu_Q\rangle\right] + 4\langle\mu_P,\mu_P\rangle\langle\mu_P,\mu_Q\rangle\\
&\quad - 4\,\mathbb{E}\left[\langle\varphi(y),\mu_Q\rangle\langle\varphi(y),\mu_P\rangle\right] + 4\langle\mu_Q,\mu_Q\rangle\langle\mu_P,\mu_Q\rangle.
\end{aligned}
$$

Estimators The various terms in ζ_1 and ζ_2 can be estimated with finite samples as follows. Define a matrix K_{XX} by $(K_{XX})_{ij} = k(X_i,X_j)$, K_{YY} by $(K_{YY})_{ij} = k(Y_i,Y_j)$, and K_{XY} by $(K_{XY})_{ij} = k(X_i,Y_j)$. Let $\tilde{K}_{XX}$ and $\tilde{K}_{YY}$ be K_{XX} and K_{YY} with their diagonals set to zero. Let e be the m-vector of all ones.

$$
\begin{aligned}
\langle\mu_P,\mu_Q\rangle &\approx \frac{1}{m^2}\sum_{i,j} k(x_i,y_j) = \frac{1}{m^2} e^\mathsf{T} K_{XY} e\\
\langle\mu_P,\mu_P\rangle &\approx \frac{1}{m(m-1)}\sum_{i\neq j} k(x_i,x_j) = \frac{1}{m(m-1)} e^\mathsf{T} \tilde{K}_{XX} e\\
\mathbb{E}\left[\langle\varphi(y),\mu_Q\rangle^2\right] &\approx \frac{1}{m}\sum_i \langle\varphi(y_i),\mu_Q\rangle^2\\
&\approx \frac{1}{m}\sum_i \frac{1}{m-1}\sum_{j\neq i}\frac{1}{m-2}\sum_{\ell\notin\{i,j\}} \langle\varphi(y_i),\varphi(y_j)\rangle\langle\varphi(y_i),\varphi(y_\ell)\rangle\\
&= \frac{1}{m(m-1)(m-2)}\sum_i\left(\sum_{j,\ell\neq i} k(y_i,y_j)k(y_i,y_\ell) - \sum_{j\neq i} k(y_i,y_j)^2\right)\\
&= \frac{1}{m(m-1)(m-2)}\sum_i\left(\sum_{j,\ell\neq i} (K_{YY})_{i,j}(K_{YY})_{i,\ell} - \sum_{j\neq i} (K_{YY})_{i,j}^2\right)\\
&= \frac{1}{m(m-1)(m-2)}\sum_i\left(\sum_{j,\ell} (\tilde{K}_{YY})_{i,j}(\tilde{K}_{YY})_{i,\ell} - \sum_{j} (\tilde{K}_{YY})_{i,j}^2\right)\\
&= \frac{1}{m(m-1)(m-2)}\left(\sum_{j,\ell} (\tilde{K}_{YY}\tilde{K}_{YY})_{j\ell} - \sum_{i,j} (\tilde{K}_{YY})_{i,j}^2\right)\\
&= \frac{1}{m(m-1)(m-2)}\left(\|\tilde{K}_{YY}e\|^2 - \|\tilde{K}_{YY}\|_F^2\right)\\
\mathbb{E}\left[\langle\varphi(x),\mu_Q\rangle^2\right] &\approx \frac{1}{m}\sum_i \langle\varphi(x_i),\mu_Q\rangle^2\\
&\approx \frac{1}{m}\sum_i \frac{1}{m}\sum_j \frac{1}{m-1}\sum_{\ell\neq j} \langle\varphi(x_i),\varphi(y_j)\rangle\langle\varphi(x_i),\varphi(y_\ell)\rangle\\
&= \frac{1}{m^2(m-1)}\sum_i\left(\sum_{j,\ell} k(x_i,y_j)k(x_i,y_\ell) - \sum_j k(x_i,y_j)^2\right)\\
&= \frac{1}{m^2(m-1)}\left(\|K_{XY}e\|^2 - \|K_{XY}\|_F^2\right)\\
\mathbb{E}\left[\langle\varphi(y),\mu_Q\rangle\langle\varphi(y),\mu_P\rangle\right] &\approx \frac{1}{m}\sum_i \left\langle \varphi(y_i), \frac{1}{m-1}\sum_{j\neq i}\varphi(y_j)\right\rangle \left\langle \varphi(y_i), \frac{1}{m}\sum_\ell \varphi(x_\ell)\right\rangle
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{m^2(m-1)} \sum_i \sum_{j \neq i} \sum_\ell k(y_i, y_j) k(y_i, x_\ell) \\
&= \frac{1}{m^2(m-1)} \sum_i \sum_j \sum_\ell (\tilde{K}_{YY})_{i,j} (K_{XY})_{\ell,i} \\
&= \frac{1}{m^2(m-1)} e^\mathsf{T} \tilde{K}_{YY} K_{XY}^\mathsf{T} e \\
\mathbb{E}[k(x_1, x_2)^2] &\approx \frac{1}{m(m-1)} \sum_{i \neq j} k(x_i, x_j)^2 = \frac{1}{m(m-1)} \|\tilde{K}_{XX}\|_F^2 \\
\mathbb{E}[k(x, y)^2] &\approx \frac{1}{m^2} \|K_{XY}\|_F^2
\end{aligned}
$$

The estimator $\hat{V}_m$ is then obtained by replacing each term of ζ_1 and ζ_2 with the appropriate estimator, above:

$$
\begin{aligned}
\widehat{V}_m := \frac{4(m-2)}{m(m-1)} \Bigg[\\
&\frac{1}{m(m-1)(m-2)} \left(\|\tilde{K}_{XX} e\|^2 - \|\tilde{K}_{XX}\|_F^2 \right) - \frac{1}{m^2(m-1)^2} \left(e^\mathsf{T} \tilde{K}_{XX} e \right)^2 \\
&+ \frac{1}{m(m-1)(m-2)} \left(\|\tilde{K}_{YY} e\|^2 - \|\tilde{K}_{YY}\|_F^2 \right) - \frac{1}{m^2(m-1)^2} \left(e^\mathsf{T} \tilde{K}_{YY} e \right)^2 \\
&+ \frac{1}{m^2(m-1)} \left(\|K_{XY} e\|^2 - \|K_{XY}\|_F^2 \right) - \frac{1}{m^4} \left(e^\mathsf{T} K_{XY} e \right)^2 \\
&+ \frac{1}{m^2(m-1)} \left(\|K_{XY}^\mathsf{T} e\|^2 - \|K_{XY}\|_F^2 \right) - \frac{1}{m^4} \left(e^\mathsf{T} K_{XY} e \right)^2 \\
&- \frac{2}{m^2(m-1)} e^\mathsf{T} \tilde{K}_{XX} K_{XY} e + \frac{2}{m^3(m-1)} \left(e^\mathsf{T} \tilde{K}_{XX} e \right) \left(e^\mathsf{T} K_{XY} e \right) \\
&- \frac{2}{m^2(m-1)} e^\mathsf{T} \tilde{K}_{YY} K_{XY}^\mathsf{T} e + \frac{2}{m^3(m-1)} \left(e^\mathsf{T} \tilde{K}_{YY} e \right) \left(e^\mathsf{T} K_{XY} e \right) \\
\Bigg] + \frac{2}{m(m-1)} \Bigg[\\
&\frac{1}{m(m-1)} \|\tilde{K}_{XX}\|_F^2 - \frac{1}{m^2(m-1)^2} \left(e^\mathsf{T} \tilde{K}_{XX} e \right)^2 \\
&+ \frac{1}{m(m-1)} \|\tilde{K}_{YY}\|_F^2 - \frac{1}{m^2(m-1)^2} \left(e^\mathsf{T} \tilde{K}_{YY} e \right)^2 \\
&+ \frac{2}{m^2} \|K_{XY}\|_F^2 - \frac{2}{m^4} \left(e^\mathsf{T} K_{XY} e \right)^2 \\
&- \frac{4}{m^2(m-1)} e^\mathsf{T} \tilde{K}_{XX} K_{XY} e + \frac{4}{m^3(m-1)} \left(e^\mathsf{T} \tilde{K}_{XX} e \right) \left(e^\mathsf{T} K_{XY} e \right) \\
&- \frac{4}{m^2(m-1)} e^\mathsf{T} \tilde{K}_{YY} K_{XY}^\mathsf{T} e + \frac{4}{m^3(m-1)} \left(e^\mathsf{T} \tilde{K}_{YY} e \right) \left(e^\mathsf{T} K_{XY} e \right) \Bigg].
\end{aligned}
$$

(5) follows by simple algebraic manipulations.