Revisiting classifier two-sample tests

David Lopez-Paz, Maxime Oquab

REVISITING CLASSIFIER TWO-SAMPLE TESTS

David Lopez-Paz[1], **Maxime Oquab**[1,2]
[1]Facebook AI Research, [2]WILLOW project team, Inria / ENS / CNRS
dlp@fb.com, maxime.oquab@inria.fr

ABSTRACT

The goal of two-sample tests is to assess whether two samples, $S_P \sim P^n$ and $S_Q \sim Q^m$, are drawn from the same distribution. Perhaps intriguingly, one relatively unexplored method to build two-sample tests is the use of binary classifiers. In particular, construct a dataset by pairing the n examples in S_P with a positive label, and by pairing the m examples in S_Q with a negative label. If the null hypothesis "$P = Q$" is true, then the classification accuracy of a binary classifier on a held-out subset of this dataset should remain near chance-level. As we will show, such *Classifier Two-Sample Tests* (C2ST) learn a suitable representation of the data on the fly, return test statistics in interpretable units, have a simple null distribution, and their predictive uncertainty allow to interpret where P and Q differ.

The goal of this paper is to establish the properties, performance, and uses of C2ST. First, we analyze their main theoretical properties. Second, we compare their performance against a variety of state-of-the-art alternatives. Third, we propose their use to evaluate the sample quality of generative models with intractable likelihoods, such as Generative Adversarial Networks (GANs). Fourth, we showcase the novel application of GANs together with C2ST for causal discovery.

1 INTRODUCTION

One of the most fundamental problems in statistics is to assess whether two samples, $S_P \sim P^n$ and $S_Q \sim Q^m$, are drawn from the same probability distribution. To this end, *two-sample tests* (**?**) summarize the differences between the two samples into a real-valued test *statistic*, and then use the value of such statistic to accept[1] or reject the null hypothesis "$P = Q$". The development of powerful two-sample tests is instrumental in a myriad of applications, including the evaluation and comparison of generative models. Over the last century, statisticians have nurtured a wide variety of two-sample tests. However, most of these tests are only applicable to one-dimensional examples, require the prescription of a fixed representation of the data, return test statistics in units that are difficult to interpret, or do not explain *how* the two samples under comparison differ.

Intriguingly, there exists a relatively unexplored strategy to build two-sample tests that overcome the aforementioned issues: training a binary classifier to distinguish between the examples in S_P and the examples in S_Q. Intuitively, if $P = Q$, the test accuracy of such binary classifier should remain near chance-level. Otherwise, if $P \neq Q$ and the binary classifier is able to unveil some of the distributional differences between S_P and S_Q, its test accuracy should depart from chance-level. As we will show, such *Classifier Two-Sample Tests* (C2ST) learn a suitable representation of the data on the fly, return test statistics in interpretable units, have simple asymptotic distributions, and their learned features and predictive uncertainty provide interpretation on *how* P and Q differ. In such a way, this work brings together the communities of statistical testing and representation learning.

The goal of this paper is to establish the theoretical properties and evaluate the practical uses of C2ST. To this end, our **contributions** are:

- We review the basics of two-sample tests in Section **??**, as well as their common applications to measure statistical dependence and evaluate generative models.
- We analyze the attractive properties of C2ST (Section **??**) including an analysis of their exact asymptotic distributions, testing power, and interpretability.

[1] For clarity, we abuse statistical language and write "accept" to mean "fail to reject".

- We evaluate C2ST on a wide variety of synthetic and real data (Section **??**), and compare their performance against multiple state-of-the-art alternatives. Furthermore, we provide examples to illustrate how C2ST can interpret the differences between pairs of samples.
- In Section **??**, we propose the use of classifier two-sample tests to evaluate the sample quality of generative models with intractable likelihoods, such as Generative Adversarial Networks (**?**), also known as GANs.
- As a novel application of the synergy between C2ST and GANs, Section **??** proposes the use of these methods for causal discovery.

2 Two-Sample Testing

The goal of two-sample tests is to assess whether two samples, denoted by $S_P \sim P^n$ and $S_Q \sim Q^m$, are drawn from the same distribution (**?**). More specifically, two-sample tests either accept or reject the *null hypothesis*, often denoted by H_0, which stands for "$P = Q$". When rejecting H_0, we say that the two-sample test favors the *alternative hypothesis*, often denoted by H_1, which stands for "$P \neq Q$". To accept or reject H_0, two-sample tests summarize the differences between the two *samples* (sets of identically and independently distributed *examples*):

$$S_P := \{x_1, \ldots, x_n\} \sim P^n(X) \text{ and } S_Q := \{y_1, \ldots, y_m\} \sim Q^m(Y) \tag{1}$$

into a statistic $\hat{t} \in \mathbb{R}$. Without loss of generality, we assume that the two-sample test returns a small statistic when the null hypothesis "$P = Q$" is true, and a large statistic otherwise. Then, for a sufficiently small statistic, the two-sample test will accept H_0. Conversely, for a sufficiently large statistic, the two-sample test will reject H_0 in favour of H_1.

More formally, the statistician performs a two-sample test in four steps. First, decide a *significance level* $\alpha \in [0, 1]$, which is an input to the two-sample test. Second, compute the two-sample test statistic $\hat{t}$. Third, compute the *p-value* $\hat{p} = P(T \geq \hat{t} | H_0)$, the probability of the two-sample test returning a statistic as large as $\hat{t}$ when H_0 is true. Fourth, reject H_0 if $\hat{p} < \alpha$, and accept it otherwise.

Inevitably, two-sample tests can fail in two different ways. First, to make a *type-I error* is to reject the null hypothesis when it is true (a "false positive"). By the definition of p-value, the probability of making a type-I error is upper-bounded by the significance level α. Second, to make a *type-II error* is to accept the null hypothesis when it is false (a "false negative"). We denote the probability of making a type-II error by β, and refer to the quantity $\pi = 1 - \beta$ as the *power* of a test. Usually, the statistician uses domain-specific knowledge to evaluate the consequences of a type-I error, and thus prescribe an appropriate significance level α. Within the prescribed significance level α, the statistician prefers the two-sample test with maximum power π.

Among others, two-sample tests serve two other uses. First, two-sample tests can *measure statistical dependence* (**?**). In particular, testing the independence null hypothesis "the random variables X and Y are independent" is testing the two-sample null hypothesis "$P(X, Y) = P(X)P(Y)$". In practice, the two-sample test would compare the sample $S = \{(x_i, y_i)\}_{i=1}^n \sim P(X, Y)^n$ to a sample $S_\sigma = \{(x_i, y_{\sigma(i)})\}_{i=1}^n \sim (P(X)P(Y))^n$, where σ is a random permutation of the set of indices $\{1, \ldots, n\}$. This approach is consistent when considering all possible random permutations. However, since independence testing is a subset of two-sample testing, specialized independence tests may exhibit higher power for this task (**?**).

Second, two-sample tests can *evaluate the sample quality of generative models* with intractable likelihoods, but tractable sampling procedures. Intuitively, a generative model produces good samples $\hat{S} = \{\hat{x}_i\}_{i=1}^n$ if these are indistinguishable from the real data $S = \{x_i\}_{i=1}^n$ that they model. Thus, the two-sample test statistic between $\hat{S}$ and S measures the fidelity of the samples $\hat{S}$ produced by the generative model. The use of two-sample tests to evaluate the sample quality of generative models include the pioneering work of **?**, the use of Maximum Mean Discrepancy (MMD) criterion (**?????**), and the connections to density-ratio estimation (**????**).

Over the last century, statisticians have nurtured a wide variety of two-sample tests. Classical two-sample tests include the t-test (**?**), which tests for the difference in means of two samples; the Wilcoxon-Mann-Whitney test (**??**), which tests for the difference in rank means of two samples; and the Kolmogorov-Smirnov tests (**??**) and their variants (**?**), which test for the difference in the empirical cumulative distributions of two samples. However, these classical tests are only efficient when applied to one-dimensional data. Recently, the use of kernel methods (**?**) enabled the development of two-sample tests applicable to multidimensional data. Examples of these tests include the MMD test (**?**), which looks for differences in the empirical kernel mean embeddings of two samples, and the Mean Embedding test or ME (**??**), which looks for differences in the empirical kernel mean embeddings of two samples at optimized locations. However, kernel two-sample tests require the prescription of a manually-engineered representation of the data under study, and return values in units that are difficult to interpret. Finally, only the ME test provides a mechanism to interpret how P and Q differ.

Next, we discuss a simple but relatively unexplored strategy to build two-sample tests that overcome these issues: the use of binary classifiers.

3 Classifier Two-Sample Tests (C2ST)

Without loss of generality, we assume access to the two samples S_P and S_Q defined in (**??**), where $x_i, y_j \in \mathcal{X}$, for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$, and $m = n$. To test whether the null hypothesis $H_0 : P = Q$ is true, we proceed in five steps. First, construct the dataset

$$\mathcal{D} = \{(x_i, 0)\}_{i=1}^n \cup \{(y_i, 1)\}_{i=1}^n =: \{(z_i, l_i)\}_{i=1}^{2n}.$$

Second, shuffle $\mathcal{D}$ at random, and split it into the disjoint *training* and testing subsets $\mathcal{D}_{\text{tr}}$ and $\mathcal{D}_{\text{te}}$, where $\mathcal{D} = \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{te}}$ and $n_{\text{te}} := |\mathcal{D}_{\text{te}}|$. Third, train a binary classifier $f : \mathcal{X} \to [0, 1]$ on $\mathcal{D}_{\text{tr}}$; in the following, we assume that $f(z_i)$ is an estimate of the conditional probability distribution $p(l_i = 1|z_i)$. Fourth, return the classification accuracy on $\mathcal{D}_{\text{te}}$:

$$\hat{t} = \frac{1}{n_{\text{te}}} \sum_{(z_i, l_i) \in \mathcal{D}_{\text{te}}} \mathbb{I}\left[\mathbb{I}\left(f(z_i) > \frac{1}{2}\right) = l_i\right] \tag{2}$$

as our *C2ST statistic*, where $\mathbb{I}$ is the indicator function. The intuition here is that if $P = Q$, the test accuracy (**??**) should remain near chance-level. In opposition, if $P \neq Q$ and the binary classifier unveils distributional differences between the two samples, the test classification accuracy (**??**) should be *greater* than chance-level. Fifth, to accept or reject the null hypothesis, compute a p-value using the null distribution of the C2ST, as discussed next.

3.1 Null and Alternative Distributions

Each term $\mathbb{I}\left[\mathbb{I}(f(z_i) > 1/2) = l_i\right]$ appearing in (**??**) is an independent Bernoulli(p_i) random variable, where p_i is the probability of classifying correctly the example z_i in $\mathcal{D}_{\text{te}}$.

First, under the null hypothesis $H_0 : P = Q$, the samples $S_P \sim P^n$ and $S_Q \sim Q^m$ follow the same distribution, leading to an impossible binary classification problem. In that case, $n_{\text{te}}\hat{t}$ follows a Binomial$(n_{\text{te}}, p = \frac{1}{2})$ distribution. Therefore, for large n_{te}, we can use the central limit theorem to approximate the null distribution of (**??**) by $\mathcal{N}(\frac{1}{2}, \frac{1}{4n_{\text{te}}})$.

Second, under the alternative hypothesis $H_1 : P \neq Q$, the statistic $n_{\text{te}}\hat{t}$ follows a Poisson Binomial distribution, since the constituent Bernoulli random variables may not be identically distributed. In the following, we will approximate such Poisson Binomial distribution by the Binomial$(n, \bar{p})$ distribution, where $\bar{p} = \frac{1}{n}\sum_{i=1}^n p_i$ (**?**). Therefore, we can use the central limit theorem to approximate the alternative distribution of (**??**) by $\mathcal{N}(\bar{p}, \frac{\bar{p}(1-\bar{p})}{n_{\text{te}}})$.

3.2 Testing power

To analyze the power (probability of correctly rejecting false null hypothesis) of C2ST, we assume that the our classifier has an expected (unknown) accuracy of $H_0 : t = \frac{1}{2}$ under the null hypothesis "$P = Q$", and an expected accuracy of $H_1 : t = \frac{1}{2} + \epsilon$ under the alternative hypothesis "$P \neq Q$", where $\epsilon \in (0, \frac{1}{2})$ is the *effect size* distinguishing P from Q. Let Φ be the Normal cdf, n_{te} the number of samples available for testing, and α the significance level. Then,

Theorem 1. *Given the conditions described in the previous paragraph, the approximate power of the statistic* (**??**) *is* $\Phi\left(\frac{\epsilon\sqrt{n_{te}} - \Phi^{-1}(1-\alpha)/2}{\sqrt{\frac{1}{4} - \epsilon^2}}\right)$.

See Appendix **??** for a proof. The power bound in Theorem **??** has an optimal order of magnitude for multi-dimensional problems (**???**). These are problems with fixed d and $n \to \infty$, where the power bounds do not depend on d.

Remark 1. *We leave for future work the study of quadratic-time C2ST with optimal power in high-dimensional problems (**?**). These are problems where the ratio $n/d \to c \in [0, 1]$, and the power bounds depend on d. One possible line of research in this direction is to investigate the power and asymptotic distributions of quadratic-time C2ST statistics $\frac{1}{n_{te}(n_{te}-1)} \sum_{i \neq j} \mathbb{I}[\mathbb{I}(f(z_i, z_j) > \frac{1}{2}) = l_i]$, where the classifier $f(z, z')$ predicts if the examples (z, z') come from the same sample.*

Theorem **??** also illustrates that maximizing the power of a C2ST is a trade-off between two competing objectives: choosing a classifier that *maximizes the test accuracy* ϵ and *maximizing the size of the test set* n_{te}. This relates to the well known bias-variance trade-off in machine learning. Indeed, simple classifiers will miss more nonlinear patterns in the data (leading to smaller test accuracy), but call for less training data (leading to larger test set sizes). On the other hand, flexible classifiers will miss less nonlinear patterns in the data (leading to higher test accuracy), but call for more training data (leading to smaller test sizes). Formally, the relationship between the test accuracy, sample size, and the flexibility of a classifier depends on capacity measures such as the VC-Dimension (**?**). Note that there is no restriction to perform model selection (such as cross-validation) on $\mathcal{D}_{\text{tr}}$.

Remark 2. *We have focused on test statistics (**??**) built on top of the zero-one loss $\ell_{0-1}(y, y') = \mathbb{I}[y = y'] \in \{0, 1\}$. These statistics give rise to Bernoulli random variables, which can exhibit high variance. However, our arguments are readily extended to real-valued binary classification losses. Then, the variance of such real-valued losses would describe the norm of the decision function of the classifier two-sample test, appear in the power expression from Theorem **??**, and serve as a hyper-parameter to maximize power as in (**?**, Section 3).*[2]

3.3 Interpretability

There are three ways to interpret the result of a C2ST. First, recall that the classifier predictions $f(z_i)$ are estimates of the conditional probabilities $p(l_i = 1|z_i)$ for each of the samples z_i in the test set. Inspecting these probabilities together with the true labels l_i determines which examples were correctly or wrongly labeled by the classifier, with the least or the most confidence. Therefore, the values $f(z_i)$ explain *where* the two distributions differ. Second, C2ST inherit the interpretability of their classifiers to explain which *features* are most important to distinguish distributions, in the same way as the ME test (**?**). Examples of interpretable features include the filters of the first layer of a neural network, the feature importance of random forests, the weights of a generalized linear model, and so on. Third, C2ST return statistics $\hat{t}$ in interpretable units: these relate to the percentage of samples correctly distinguishable between the two distributions. These interpretable numbers can complement the use of p-values.

3.4 Prior Uses

The reduction of two-sample testing to binary classification was introduced in (**?**), studied within the context of information theory in (**??**), discussed in (**??**), and analyzed (for the case of linear discriminant analysis) in (**?**). The use of binary classifiers for two-sample testing is increasingly common in neuroscience: see (**??**) and the references therein. Implicitly, binary classifiers also perform two-sample tests in algorithms that discriminate data from noise, such as unsupervised-as-supervised learning (**?**), noise contrastive estimation (**?**), negative sampling (**?**), and GANs (**?**).

4 Experiments on Two-Sample Testing

We study two variants of classifier-based two-sample tests (C2ST): one based on neural networks (C2ST-NN), and one based on k-nearest neighbours (C2ST-KNN). C2ST-NN has one hidden layer of 20 ReLU neurons, and trains for 100 epochs using the Adam optimizer (**?**). C2ST-KNN uses $k = \lfloor n_{\text{tr}}^{1/2} \rfloor$ nearest neighbours for classification. Throughout our experiments, we did not observe

[2] For a related discussion on this issue, we recommend the insightful comment by Arthur Gretton and Wittawat Jitkrittum, available at `https://openreview.net/forum?id=SJkXfE5xx`.

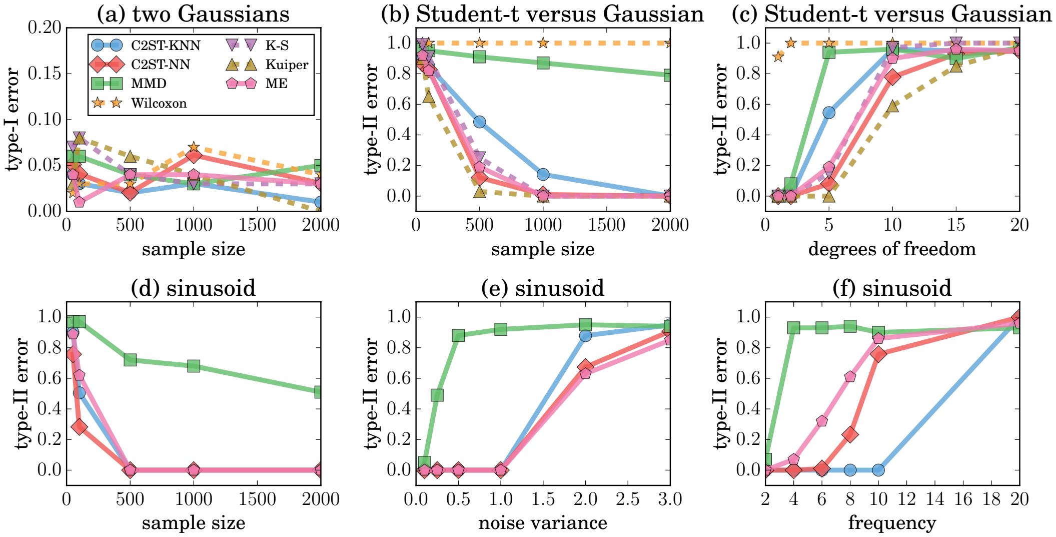


Figure 1: Results (type-I and type-II errors) of our synthetic two-sample test experiments.

a significant improvement in performance when increasing the flexibility of these classifiers (e.g., increasing the number of hidden neurons or decreasing the number of nearest neighbors). When analyzing one-dimensional data, we compare the performance of C2ST-NN and C2ST-KNN against the Wilcoxon-Mann-Whitney test (**??**), the Kolmogorov-Smirnov test (**??**), and the Kuiper test (**?**). In all cases, we also compare the performance of C2ST-NN and C2ST-KNN against the linear-time estimate of the Maximum Mean Discrepancy (MMD) criterion (**?**), the ME test (**?**), and the SCF test (**?**). We use a significance level $\alpha = 0.05$ across all experiments and tests, unless stated otherwise. We use Gaussian approximations to compute the null distributions of C2ST-NN and C2ST-KNN. We use the implementations of the MMD, ME, and SCF tests gracefully provided by **?**, the scikit-learn implementation of the Kolmogorov-Smirnov and Wilcoxon tests, and the implementation from `https://github.com/aarchiba/kuiper` of the Kuiper test. The implementation of our experiments is available at `https://github.com/lopezpaz/classifier_tests`.

4.1 Experiments on Two-Sample Testing

Control of type-I errors We start by evaluating the correctness of all the considered two-sample tests by examining if the prescribed significance level $\alpha = 0.05$ upper-bounds their type-I error. To do so, we draw $x_1, \ldots, x_n, y_1, \ldots, y_n \sim \mathcal{N}(0, 1)$, and run each two-sample test on the two samples $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$. In this setup, a type-I error would be to reject the true null hypothesis. Figure **??**(a) shows that the type-I error of all tests is upper-bounded by the prescribed significance level, for all $n \in \{25, 50, 100, 500, 1000, 5000, 10000\}$ and 100 random repetitions. Thus, all tests control their type-I error as expected, up to random variations due to finite experiments.

Gaussian versus Student We consider distinguishing between samples drawn from a Normal distribution and samples drawn from a Student's t-distribution with ν degrees of freedom. We shift and scale both samples to exhibit zero-mean and unit-variance. Since the Student's t distribution approaches the Normal distribution as ν increases, a two-sample test must focus on the peaks of the distributions to distinguish one from another. Figure **??**(b,c) shows the percentage of type-II errors made by all tests as we vary separately n and ν, over 100 trials (random samples). We set $n = 2000$ when ν varies, and let $\nu = 3$ when n varies. The Wilcoxon-Mann-Whitney exhibits the worst performance, as expected (since the ranks mean of the Gaussian and Student's t distributions coincide) in this experiment. The best performing method is the the one-dimensional Kuiper test, followed closely by the multi-dimensional tests C2ST-NN and ME.

Independence testing on sinusoids For completeness, we showcase the use two-sample tests to measure statistical dependence. This can be done, as described in Section **??**, by performing a two-sample test between the observed data $\{(x_i, y_i)\}_{i=1}^n$ and $\{(x_i, y_{\sigma(i)})\}_{i=1}^n$, where σ is a random

Problem	n^{te}	ME-full	ME-grid	SCF-full	SCF-grid	MMD-quad	MMD-lin	C2ST-NN
Bayes-Bayes	215	.012	.018	.012	.004	.022	.008	**.002**
Bayes-Deep	216	.954	.034	.688	.180	.906	.262	**1.00**
Bayes-Learn	138	.990	.774	.836	.534	**1.00**	.238	**1.00**
Bayes-Neuro	394	**1.00**	.300	.828	.500	.952	.972	**1.00**
Learn-Deep	149	.956	.052	.656	.138	.876	.500	**1.00**
Learn-Neuro	146	.960	.572	.590	.360	**1.00**	.538	**1.00**

Table 1: Type-I errors (first row) and powers (rest of rows) in distinguishing NIPS papers categories.

Problem	n^{te}	ME-full	ME-grid	SCF-full	SCF-grid	MMD-quad	MMD-lin	C2ST-NN
$\pm$ vs. $\pm$	201	.010	.012	.014	**.002**	.018	.008	**.002**
$+$ vs. $-$	201	.998	.656	**1.00**	.750	**1.00**	.578	.997

Table 2: Type-I errors (first row) and powers (second row) in distinguishing facial expressions.

permutation. Since the distributions $P(X)P(Y)$ and $P(X, Y)$ are bivariate, only the C2ST-NN, C2ST-KNN, MMD, and ME tests compete in this task. We draw (x_i, y_i) according to the generative model $x_i \sim \mathcal{N}(0, 1)$, $\epsilon_i \sim \mathcal{N}(0, \gamma^2)$, and $y_i \sim \cos(\delta x_i) + \epsilon_i$. Here, x_i are iid examples from the random variable X, and y_i are iid examples from the random variable Y. Thus, the statistical dependence between X and Y weakens as we increase the frequency δ of the sinusoid, or increase the variance γ^2 of the additive noise. Figure **??**(d,e,f) shows the percentage of type-II errors made by C2ST-NN, C2ST-KNN, MMD, and ME as we vary separately n, δ, and γ over 100 trials. We let $n = 2000$, $\delta = 1$, $\gamma = 0.25$ when fixed. Figure **??**(d,e,f) reveals that among all tests, C2ST-NN is the most efficient in terms of sample size, C2ST-KNN is the most robust with respect to high-frequency variations, and that C2ST-NN and ME are the most robust with respect to additive noise.

Distinguishing between NIPS articles We consider the problem of distinguishing between some of the categories of the 5903 articles published in the Neural Information Processing Systems (NIPS) conference from 1988 to 2015, as discussed in **?**. We consider articles on Bayesian inference (Bayes), neuroscience (Neuro), deep learning (Deep), and statistical learning theory (Learn). Table **??** shows the type-I errors (Bayes-Bayes row) and powers (rest of rows) for the tests reported in (**?**), together with C2ST-NN, at a significance level $\alpha = 0.01$, when averaged over 500 trials. In these experiments, C2ST-NN achieves maximum power, while upper-bounding its type-I error by α.

Distinguishing between facial expressions Finally, we apply C2ST-NN to the problem of distinguishing between positive (happy, neutral, surprised) and negative (afraid, angry, disgusted) facial expressions from the Karolinska Directed Emotional Faces dataset, as discussed in (**?**). See the fourth plot of Figure **??**, first two-rows, for one example of each of these six emotions. Table **??** shows the type-I errors ($\pm$ vs $\pm$ row) and the powers ($+$ vs $-$ row) for the tests reported in (**?**), together with C2ST-NN, at $\alpha = 0.01$, averaged over 500 trials. C2ST-NN achieves a near-optimal power, only marginally behind the perfect results of SCF-full and MMD-quad.

5 Experiments on Generative Adversarial Network Evaluation

Since effective generative models will produce examples barely distinguishable from real data, two-sample tests arise as a natural alternative to evaluate generative models. Particularly, our interest is to evaluate the sample quality of generative models with intractable likelihoods, such as GANs (**?**). GANs implement the adversarial game

$$\min_g \max_d \mathop{\mathbb{E}}_{x \sim P(X)} [\log(d(x))] + \mathop{\mathbb{E}}_{z \sim P(Z)} [\log(1 - d(g(z)))] , \tag{3}$$

where $d(x)$ depicts the probability of the example x following the data distribution $P(X)$ versus being synthesized by the generator. This is according to a trainable *discriminator* function d. In the adversarial game, the generator g plays to fool the discriminator d by transforming noise vectors $z \sim P(Z)$ into real-looking examples $g(z)$. On the opposite side, the discriminator plays to distinguish between real examples x and synthesized examples $g(z)$. To approximate the solution to (**??**), alternate the optimization of the two losses (**?**) given by

$$\begin{aligned} L_d(d) &= \mathbb{E}_x \left[\ell(d(x), 1)\right] + \mathbb{E}_z \left[\ell(d(g(z)), 0)\right], \\ L_g(g) &= \mathbb{E}_x \left[\ell(d(x), 0)\right] + \mathbb{E}_z \left[\ell(d(g(z)), 1)\right]. \end{aligned} \tag{4}$$

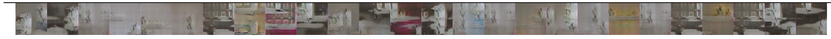
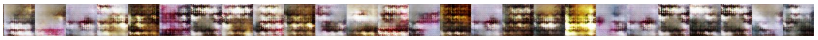
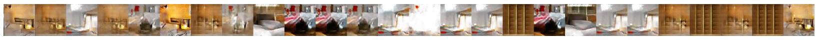
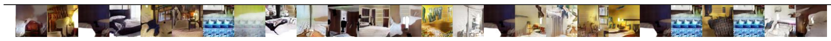
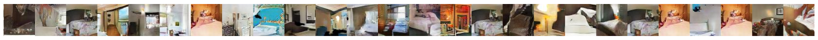
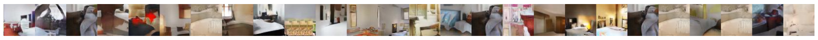
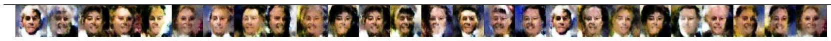
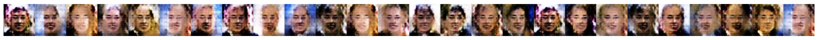
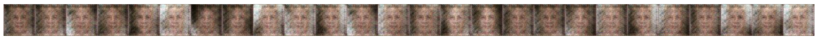
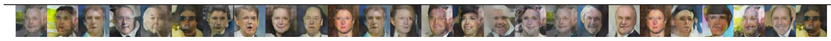
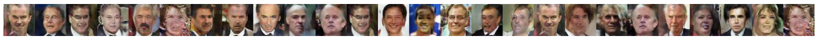
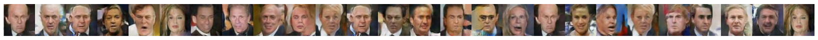
random sample	MMD	KNN	NN
	0.158	0.830	0.999
	0.154	0.994	1.000
	0.048	0.962	1.000
	**0.012**	0.798	0.964
	0.024	0.748	**0.949**
	0.019	**0.670**	0.983
	0.152	0.940	1.000
	0.222	0.978	1.000
	0.715	1.000	1.000
	**0.015**	0.817	0.987
	0.020	0.784	**0.950**
	0.024	**0.697**	0.971

Table 3: Results on GAN evaluation. Lower test statistics are best. Full results in Appendix **??**.

Under the formalization (**??**), the adversarial game reduces to the sequential minimization of $L_d(d)$ and $L_g(g)$, and reveals the true goal of the discriminator: to be the C2ST that best distinguishes data examples $x \sim P$ and synthesized examples $\hat{x} \sim \hat{P}$, where $\hat{P}$ is the probability distribution induced by sampling $z \sim P(Z)$ and computing $\hat{x} = g(z)$. The formalization (**??**) unveils the existence of an arbitrary binary classification loss function ℓ (See Remark **??**), which in turn decides the divergence minimized between the real and fake data distributions (**?**).

Unfortunately, the evaluation of the log-likelihood of a GANs is intractable. Therefore, we will employ a two-sample test to evaluate the quality of the fake examples $\hat{x} = g(z)$. In simple terms, evaluating a GAN in this manner amounts to withhold some real data from the training process, and use it later in a two-sample test against the same amount of synthesized data. When the two-sample test is a binary classifier (as discussed in Section **??**), this procedure is simply *training a fresh discriminator on a fresh set of data*. Since we train and test this *fresh* discriminator on held-out examples, it may differ from the discriminator trained along the GAN. In particular, the discriminator trained along with the GAN may have over-fitted to particular artifacts produced by the generator, thus becoming a poor C2ST.

We evaluate the use of two-sample tests for model selection in GANs. To this end, we train a number of DCGANs (**?**) on the bedroom class of LSUN (**?**) and the Labeled Faces in the Wild (LFW) dataset (**?**). We reused the Torch7 code of **?** to train a set of DCGANs for $\{1, 10, 50, 100, 200\}$ epochs, where the generator and discriminator networks are convolutional neural networks (**?**) with $\{1, 2, 4, 8\} \times$ gf and $\{1, 2, 4, 8\} \times$ df filters per layer, respectively. We evaluate each DCGAN on $10,000$ held-out examples using the fastest multi-dimensional two-sample tests: MMD, C2ST-NN, and C2ST-KNN.

Our first experiments revealed an interesting result. When performing two-sample tests directly on pixels, all tests obtain near-perfect test accuracy when distinguishing between real and synthesized (fake) examples. Such near-perfect accuracy happens consistently across DCGANs, regardless of the visual quality of their examples. This is because, albeit visually appealing, the fake examples contain checkerboard-like artifacts that are sufficient for the tests to consistently differentiate between real and fake examples. **?** discovered this phenomenon concurrently with us.

On a second series of experiments, we featurize all images (both real and fake) using a deep convolutional ResNet (**?**) pre-trained on ImageNet, a large dataset of natural images (**?**). In particular, we use the `resnet-34` model from **?**. Reusing a model pre-trained on natural images ensures that the test will distinguish between real and fake examples based only on natural image statistics, such as Gabor filters, edge detectors, and so on. Such a strategy is similar to perceptual losses (**?**) and inception scores (**?**). In short, in order to evaluate how natural the images synthesized by a DCGAN look, one must employ a "natural discriminator". Table **??** shows three GANs producing poor samples and three GANs producing good samples for the LSUN and LFW datasets, according to the MMD, C2ST-KNN, C2ST-NN tests on top of ResNet features. See Appendix **??** for the full

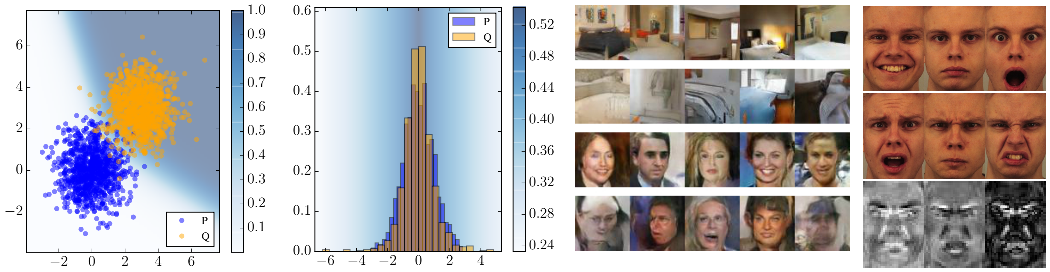


Figure 2: Interpretability of C2ST. The color map corresponds to the value of $p(l = 1|z)$.

list of results. Although it is challenging to provide with an objective evaluation of our results, we believe that the rankings provided by two-sample tests could serve for efficient early stopping and model selection.

Remark 3 (How good is my GAN? Is it overfitting?)**.** *Evaluating generative models is a delicate issue (**?**), but two-sample tests may offer some guidance. In particular, good (non-overfitting) generative models should produce similar two-sample test statistics when comparing their generated samples to both the train-set and the test-set samples.* [3] *As a general recipe, prefer* the smallest *(in number of parameters) generative model that achieves the* same and small *two-sample test statistic when comparing their generated samples to both the train-set and test-set samples.*

We have seen that GANs of different quality may lead to the same (perfect) C2ST statistic. To allow a finer comparison between generative models, we recommend implementing C2ST using a margin classifier with finite norm, or using as statistic the whole area under the C2ST training curve (on train-set or test-set samples).

5.1 Experiments on Interpretability

We illustrate the interpretability power of C2ST. First, the predictive uncertainty of C2ST sheds light on where the two samples under consideration agree or differ. In the context of GANs, this interpretability is useful to locate captured or dropped modes. In the first plot of Figure **??**, a C2ST-NN separates two bivariate Gaussian distributions with different means. When performing this separation, the C2ST-NN provides an explicit decision boundary that illustrates *where* the two distributions separate from each other. In the second plot of Figure **??**, a C2ST-NN separates a Gaussian distribution from a Student's t distribution with $\nu = 3$, after scaling both to zero-mean and unit-variance. The plot reveals that the peaks of the distributions are their most differentiating feature. Finally, the third plot of Figure **??** displays, for the LFW and LSUN datasets, five examples classified as real with high uncertainty (first row, better looking examples), and five examples classified as fake with high certainty (second row, worse looking examples).

Second, the features learnt by the classifier of a C2ST are also a mechanism to understand the differences between the two samples under study. The third plot of Figure **??** shows six examples from the Karolinska Directed Emotional Faces dataset, analyzed in Section **??**. In that same figure, we arrange the weights of the first linear layer of C2ST-NN into the feature most activated at positive examples (bottom left, positive facial expressions), the feature most activated at negative examples (bottom middle, negative facial expressions), and the "discriminative feature", obtained by substracting these two features (bottom right). The discriminative feature of C2ST-NN agrees with the one found by (**?**): positive and negative facial expressions are best distinguished at the eyebrows, smile lines, and lips. A similar analysis **?** on the C2ST-NN features in the NIPS article classification problem (Section **??**) reveals that the features most activated for the "statistical learning theory" category are those associated to the words *inequ*, *tight*, *power*, *sign*, *hypothesi*, *norm*, *hilbert*. The features most activated for the "Bayesian inference" category are those associated to the words *infer*, *markov*, *graphic*, *conjug*, *carlo*, *automat*, *laplac*.

[3] As discussed with Arthur Gretton, if the generative model memorizes the train-set samples, a sufficiently large set of generated samples would reveal such memorization to the two-sample test. This is because some unique samples would appear multiple times in the set of generated samples, but not in the test-set of samples.

Method	ANM-HSIC	IGCI	RCC	CGAN-C2ST	Ensemble	C2ST type
Accuracy	67%	71%	76%	73%	**82%**	KNN
				70%	73%	NN
				58%	65%	MMD

Table 4: Results on cause-effect discovery on the Tübingen pairs experiment.

6 Experiments on Conditional GANs for Causal Discovery

In causal discovery, we study the causal structure underlying a set of d random variables $X_1, \ldots, X_d$. In particular, we assume that the random variables $X_1, \ldots, X_d$ share a causal structure described by a collection of Structural Equations, or SEs (**?**). More specifically, we assume that the random variable X_i takes values as described by the SE $X_i = g_i(\text{Pa}(X_i, \mathcal{G}), N_i)$, for all $i = 1, \ldots, d$. In the previous, $\mathcal{G}$ is a Directed Acyclic Graph (DAG) with vertices associated to each of the random variables $X_1, \ldots, X_d$. Also in the same equation, $\text{Pa}(X_i, \mathcal{G})$ denotes the set of random variables which are parents of X_i in the graph $\mathcal{G}$, and N_i is an independent noise random variable that follows the probability distribution $P(N_i)$. Then, we say that $X_i \to X_j$ if $X_i \in \text{Pa}(X_j)$, since a change in X_i will *cause* a change in X_j, as described by the i-th SE.

The goal of causal discovery is to infer the causal graph $\mathcal{G}$ given a sample from $P(X_1, \ldots, X_d)$. For the sake of simplicity, we focus on the discovery of causal relations between two random variables, denoted by X and Y. That is, given the sample $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \sim P^n(X, Y)$, our goal is to conclude whether "X causes Y", or "Y causes X". We call this problem *cause-effect discovery* (**?**). In the case where $X \to Y$, we can write the cause-effect relationship as:

$$x \sim P(X), \quad n \sim P(N), \quad y \leftarrow g(x, n). \tag{5}$$

The current state-of-the-art in the cause-effect discovery is the family of Additive Noise Models, or ANM (**?**). These methods assume that the SE (**??**) allow the expression $y \leftarrow g(x) + n$, and exploit the independence assumption between the cause random variable X and the noise random variable N to analyze the distribution of nonlinear regression residuals, in both causal directions.

Unfortunately, assuming independent additive noise is often too simplistic (for instance, the noise could be heteroskedastic or multiplicative). Because of this reason, we propose to use Conditional Generative Adversarial Networks, or CGANs (**?**) to address the problem of cause-effect discovery. Our motivation is the shocking resemblance between the generator of a CGAN and the SE (**??**): the random variable X is the conditioning variable input to the generator, the random variable N is the noise variable input to the generator, and the random variable Y is the variable synthesized by the generator. Furthermore, CGANs respect the independence between the cause X and the noise N by construction, since $n \sim P(N)$ is independent from all other variables. This way, CGANs bypass the additive noise assumption naturally, and allow arbitrary interactions $g(X, N)$ between the cause variable X and the noise variable N.

To implement our cause-effect inference algorithm in practice, recall that training a CGAN from X to Y minimizes the two following objectives in alternation:

$$\begin{aligned} L_d(d) &= \mathbb{E}_{x,y}\ [\ell(d(x,y),1)] + \mathbb{E}_{x,z}\ [\ell(d(x,g(x,z)),0)]\,, \\ L_g(g) &= \mathbb{E}_{x,y}\ [\ell(d(x,y),0)] + \mathbb{E}_{x,z}\ [\ell(d(x,g(x,z)),1)]\,. \end{aligned}$$

Our recipe for cause-effect is to learn two CGANs: one with a generator g_y from X to Y to synthesize the dataset $\mathcal{D}_{X \to Y} = \{(x_i, g_y(x_i, z_i))\}_{i=1}^n$, and one with a generator g_x from Y to X to synthesize the dataset $\mathcal{D}_{X \leftarrow Y} = \{(g_x(y_i, z_i), y_i)\}_{i=1}^n$. Then, we prefer the causal direction $X \to Y$ if the two-sample test statistic between the real sample $\mathcal{D}$ and $\mathcal{D}_{X \to Y}$ is smaller than the one between $\mathcal{D}$ and $\mathcal{D}_{Y \to X}$. Thus, our method is Occam's razor at play: declare the simplest direction (in terms of conditional generative modeling) as the true causal direction.

Table **??** summarizes the performance of this procedure when applied to the 99 Tübingen cause-effect pairs dataset, version August 2016 (**?**). RCC is the Randomized Causation Coefficient of (**?**). The Ensemble-CGAN-C2ST trains 100 CGANs, and decides the causal direction by comparing the top generator obtained in each causal direction, as told by C2ST-KNN. The need to ensemble is a remainder of the unstable behaviour of generative adversarial training, but also highlights the promise of such models for causal discovery.

7 Conclusion

Our *take-home message* is that modern binary classifiers can be easily turned into powerful two-sample tests. We have shown that these *classifier two-sample tests* set a new state-of-the-art in performance, and enjoy unique attractive properties: they are easy to implement, learn a representation of the data on the fly, have simple asymptotic distributions, and allow different ways to interpret how the two samples under study differ. Looking into the future, the use of binary classifiers as two-sample tests provides a flexible and scalable approach for the evaluation and comparison of generative models (such as GANs), and opens the door to novel applications of these methods, such as causal discovery.

A Results on Evaluation of Generative Adversarial Networks

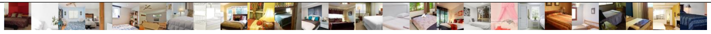
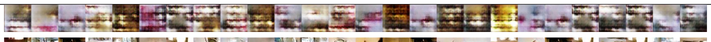
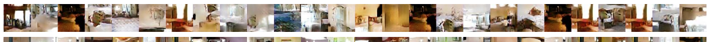
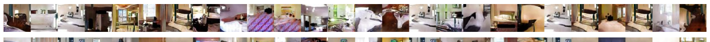
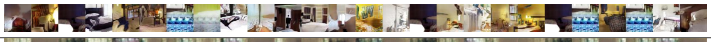
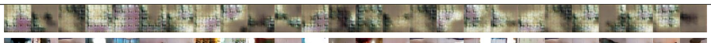
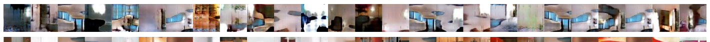
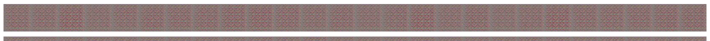
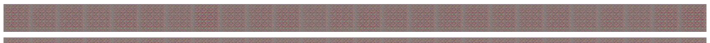
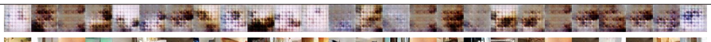
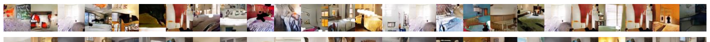
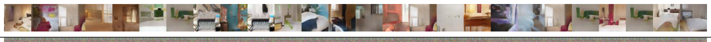
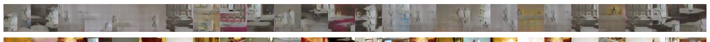
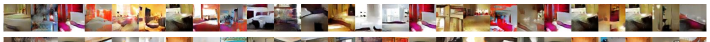
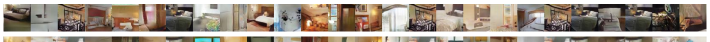
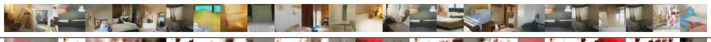
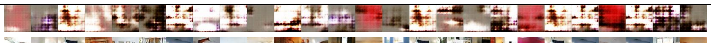
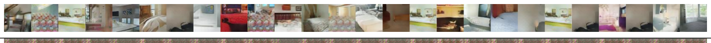
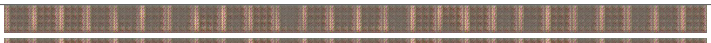
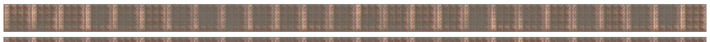
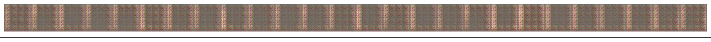
gf	df	ep	random sample	MMD	KNN	NN
-	-	-		-	-	-
32	32	1		0.154	0.994	1.000
32	32	10		0.024	0.831	0.996
32	32	50		0.026	0.758	0.983
32	32	100		0.014	0.797	0.974
32	32	200		**0.012**	0.798	0.964
32	64	1		0.330	0.984	1.000
32	64	10		0.035	0.897	0.997
32	64	50		0.020	0.804	0.989
32	64	100		0.032	0.936	0.998
32	64	200		0.048	0.962	1.000
32	96	1		0.915	0.997	1.000
32	96	10		0.927	0.991	1.000
32	96	50		0.924	0.991	1.000
32	96	100		0.928	0.991	1.000
32	96	200		0.928	0.991	1.000
64	32	1		0.389	0.987	1.000
64	32	10		0.023	0.842	0.979
64	32	50		0.018	0.788	0.977
64	32	100		0.017	0.753	0.959
64	32	200		0.018	0.736	0.963
64	64	1		0.313	0.964	1.000
64	64	10		0.021	0.825	0.988
64	64	50		0.014	0.864	0.978
64	64	100		0.019	0.685	0.978
64	64	200		0.021	0.775	0.980
64	96	1		0.891	0.996	1.000
64	96	10		0.158	0.830	0.999
64	96	50		0.015	0.801	0.980
64	96	100		0.016	0.866	0.976
64	96	200		0.020	0.755	0.983
96	32	1		0.356	0.986	1.000
96	32	10		0.022	0.770	0.991
96	32	50		0.024	0.748	**0.949**
96	32	100		0.022	0.745	0.965
96	32	200		0.024	0.689	0.981
96	64	1		0.287	0.978	1.000
96	64	10		**0.012**	0.825	**0.966**
96	64	50		0.017	0.812	0.962
96	64	100		0.019	**0.670**	0.983
96	64	200		0.020	0.711	0.972
96	96	1		0.672	0.999	1.000
96	96	10		0.671	0.999	1.000
96	96	50		0.829	0.999	1.000
96	96	100		0.668	0.999	1.000
96	96	200		0.849	0.999	1.000

Table 5: GAN evaluation results on the LSUN dataset, for all epochs (ep), filters in discriminator (df), filters in generator (gf), and test statistics (for MMD, C2ST-KNN, C2ST-NN). A lower test statistic estimates that the GAN produces better samples. Best viewed with zoom.

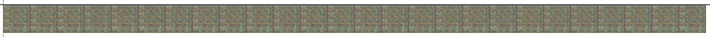
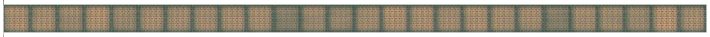
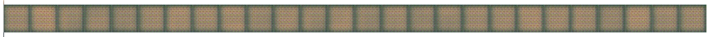
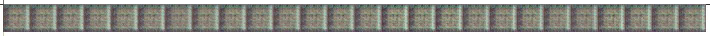
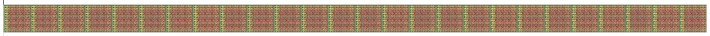
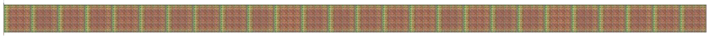
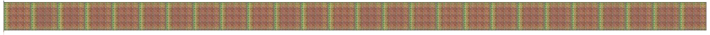
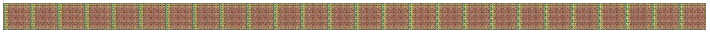
gf	df	ep	random sample	MMD	KNN	NN
-	-	-		-	-	-
32	32	1		0.806	1.000	1.000
32	32	10		0.152	0.940	1.000
32	32	50		0.042	0.788	0.993
32	32	100		0.029	0.808	0.982
32	32	200		0.022	0.776	0.970
32	64	1		0.994	1.000	1.000
32	64	10		0.989	1.000	1.000
32	64	50		0.050	0.808	0.985
32	64	100		0.036	0.766	0.972
32	64	200		**0.015**	0.817	0.987
32	96	1		0.995	1.000	1.000
32	96	10		0.992	1.000	1.000
32	96	50		0.995	1.000	1.000
32	96	100		0.053	0.778	0.987
64	96	200		0.037	0.779	0.995
64	32	1		1.041	1.000	1.000
64	32	10		0.086	0.971	1.000
64	32	50		0.043	0.756	0.988
64	32	100		0.018	0.746	0.973
64	32	200		0.025	0.757	0.972
64	64	1		0.836	1.000	1.000
64	64	10		0.103	0.910	0.998
64	64	50		0.018	0.712	0.973
64	64	100		0.020	0.784	**0.950**
64	64	200		0.022	0.719	0.974
64	96	1		1.003	1.000	1.000
64	96	10		1.015	1.000	1.000
64	96	50		1.002	1.000	1.000
64	96	100		1.063	1.000	1.000
64	96	200		1.061	1.000	1.000
96	32	1		1.022	1.000	1.000
96	32	10		0.222	0.978	1.000
96	32	50		0.026	0.734	0.965
96	32	100		0.016	0.735	0.964
96	32	200		0.021	0.780	0.973
96	64	1		0.715	1.000	1.000
96	64	10		0.042	0.904	0.999
96	64	50		0.024	**0.697**	0.971
96	64	100		0.028	0.744	0.983
96	64	200		0.020	0.697	0.976
96	96	1		0.969	1.000	1.000
96	96	10		0.920	1.000	1.000
96	96	50		0.926	1.000	1.000
96	96	100		0.920	1.000	1.000
96	96	200		0.923	1.000	1.000

Table 6: GAN evaluation results on the LFW dataset, for all epochs (ep), filters in discriminator (df), filters in generator (gf), and test statistics (for MMD, C2ST-KNN, C2ST-NN). A lower test statistic estimates that the GAN produces better samples. Best viewed with zoom.

B Proof of Theorem ??

Our statistic is a random variable $T \sim \mathcal{N}\left(\frac{1}{2}, \frac{1}{4n_{\text{te}}}\right)$ under the null hypothesis, and $T \sim \mathcal{N}\left(\frac{1}{2} + \epsilon, n_{\text{te}}^{-1}\left(\frac{1}{4} - \epsilon^2\right)\right)$ under the alternative hypothesis. Furthermore, at a significance level α, the threshold of our statistic is $z_\alpha = \frac{1}{2} + \frac{\Phi^{-1}(1-\alpha)}{\sqrt{4n_{\text{te}}}}$; under this threshold we would accept the null hypothesis. Then, the probability of making a type-II error is

$$
\begin{aligned}
\mathbb{P}_{T \sim \mathcal{N}\left(\frac{1}{2}+\epsilon, \frac{\frac{1}{4}-\epsilon^2}{n_{\text{te}}}\right)}(T < z_\alpha) &= \mathbb{P}_{T' \sim \mathcal{N}\left(0, \frac{\frac{1}{4}-\epsilon^2}{n_{\text{te}}}\right)}\left(T' < \frac{\Phi^{-1}(1-\alpha)}{\sqrt{4n_{\text{te}}}} - \epsilon\right) \\
&= \Phi\left(\sqrt{\frac{n_{\text{te}}}{\frac{1}{4}-\epsilon^2}}\left(\frac{\Phi^{-1}(1-\alpha)}{\sqrt{4n_{\text{te}}}} - \epsilon\right)\right) \\
&= \Phi\left(\frac{\Phi^{-1}(1-\alpha)/2 - \epsilon\sqrt{n_{\text{te}}}}{\sqrt{\frac{1}{4}-\epsilon^2}}\right).
\end{aligned}
$$

Therefore, the power of the test is

$$
\pi(\alpha, n_{\text{te}}, \epsilon) = 1 - \Phi\left(\frac{\Phi^{-1}(1-\alpha)/2 - \epsilon\sqrt{n_{\text{te}}}}{\sqrt{\frac{1}{4}-\epsilon^2}}\right) = \Phi\left(\frac{\epsilon\sqrt{n_{\text{te}}} - \Phi^{-1}(1-\alpha)/2}{\sqrt{\frac{1}{4}-\epsilon^2}}\right),
$$

which concludes the proof.

C Acknowledgements

We are thankful to L. Bottou, B. Graham, D. Kiela, M. Rojas-Carulla, I. Tolstikhin, and M. Tygert for their help in improving the quality of this manuscript. This work was partly supported by ERC grant LEAP (no. 336845) and CIFAR Learning in Machines & Brains program.