A Practical Guide to Sample-based Statistical Distances for Evaluating Generative Models in Science

Anonymous authors
Paper under double-blind review

Abstract

Generative models are invaluable in many fields of science because of their ability to capture high-dimensional and complicated distributions, such as photo-realistic images, protein structures, and connectomes. How do we evaluate the samples these models generate? This work aims to provide an accessible entry point to understanding popular ~~notions of~~ sample-based statistical distances, requiring only foundational knowledge in mathematics and statistics. We focus on four commonly used notions of statistical distances representing different methodologies: Using low-dimensional projections (Sliced-Wasserstein; SW), obtaining a distance using classifiers (Classifier Two-Sample Tests; C2ST), using embeddings through kernels (Maximum Mean Discrepancy; MMD), or neural networks (Fréchet Inception Distance; FID). We highlight the intuition behind each distance and explain their merits, scalability, complexity, and pitfalls. To demonstrate how these distances are used in practice, we evaluate generative models from different scientific domains, namely a model of decision making and a model generating medical images. We showcase that distinct distances can give different results on similar data. Through this guide, we aim to help researchers to use, interpret, and evaluate statistical distances for generative models in science.

1 Introduction

Generative models that produce samples of complex, high-dimensional data, have recently come to the forefront of public awareness due to their utility in a variety of scientific, clinical, engineering, and commercial domains (Bond-Taylor et al., 2021). Prominent examples include StableDiffusion (SD) and DALL-E for generating photo-realistic images (Rombach et al., 2022a), WaveNet (Oord et al., 2016) for audio synthesis, and Generative Pre-trained Transformer (GPT; Radford et al. 2018; 2019; Brown et al. 2020) for text generation. Besides this new wave of generative models, different scientific disciplines have a long history of building data generating models which capture specific processes. ~~For example in neuroscience,~~ In neuroscience, for example, the occurrence of action potentials is modeled at all different levels of detail (e.g., single neuron voltage dynamics (Hodgkin & Huxley, 1952) or at a phenomenological level (Pillow et al., 2008)), whereas in e.g., astrophysics there exist various models to simulate galaxy formation (Somerville & Davé, 2015). Along with generating novel synthetic samples, generative models can be leveraged for specific tasks, such as sample generation conditioned on class labels (e.g., diseased vs. healthy brain scans, molecules that can or cannot be synthesized; Urbina et al. 2022, class-conditional image generation; van den Oord et al. 2016; Dockhorn et al. 2022, forecasting future states of a dynamical system; Durstewitz et al. 2023; Jacobs et al. 2023; Brenner et al. 2022), data imputation (e.g., Vetter et al. 2023; Lugmayr et al. 2022), data augmentation for downstream tasks (Rommel et al., 2022), and many more (see also Table S1).

These powerful capabilities are enabled by the premise that generative models ~~accurately learn a~~ can produce samples from the high-dimensional distribution from which we assume our dataset was sampled. The dimensions can correspond to anything from individual pixels or graphs to arbitrary features of physical or abstract objects. When aiming to build generative models that better capture the true underlying data distribution, we need to answer a key question: *How accurately ~~does~~ do samples from our generative model mimic those from the true data distribution*?

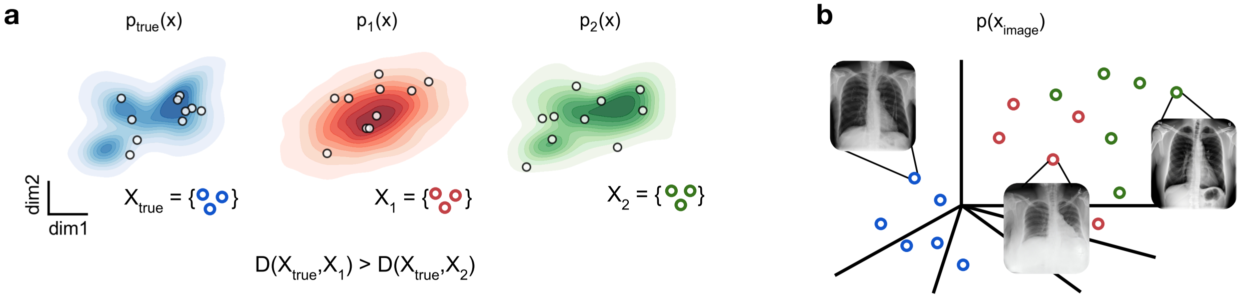

Figure 1: **The need for statistical distances in scientific generative modeling. (a)** An example target distribution, $p_{true}(x)$, and two learned distributions ($p_1(x)$ and $p_2(x)$) of different models trained to capture $p_{true}(x)$. All three distributions share the same mean and marginal variances, despite having distinct shapes. However, an appropriate sample-based distribution distance D can determine that $p_2(x)$ is more similar to $p_{true}(x)$. **(b)** Scientific applications often require evaluating high-dimensional distributions, such as distributions for images or tabular data. In this example, each point represents an X-ray image, where each dimension is one pixel.

Manual inspection of generated samples can be a good first check, e.g., in image or audio generation, where we can directly assess the visual likeness or sound quality of the samples (Gerhard et al., 2013; Vallez et al., 2022; Jayasumana et al., 2023). In general, however, we would like ~~to have quantitative distances to measure~~ quantitatively compare the similarity of distributions, for instance to ~~compare and benchmark different~~ benchmark different generative models. Many measures have been proposed that ~~provide a quantitative way to~~ assess the similarity of two distributions based on various aspects of their moments ~~, spread, central tendency, and~~ or the overall probability density (Fig. 1). Some of these measures require likelihood evaluations, as is possible with generative models such as Gaussian Mixture Models, Normalizing Flows, Variational Autoencoders, autoregressive models or diffusion models (Bishop, 2006; Papamakarios et al., 2021; Box et al., 2015; Kingma & Welling, 2014; Yenduri et al., 2023; Ho et al., 2020; Song et al., 2021). However, many contemporary machine learning models (e.g., Generative Adversarial Networks and Energy-Based Models; Goodfellow et al. 2014; Rezende et al. 2014; Hinton et al. 2006) and scientific simulators (e.g., single neuron voltage dynamics; Hodgkin & Huxley 1952) only define the likelihood *implicitly*, i.e., we can not explicitly evaluate their likelihood. ~~In this work we focus on~~ Statistical distances that can be ~~applied to~~ computed based on samples only are therefore invaluable for comparing both classes of ~~models, i. e. distances that can be computed only based on generated samples.~~ generative models (and to real data) in scientific contexts.

~~Here, we provide a guide to understanding commonly used sample-based statistical distances. Note that with *distance*, we do not necessarily refer to a distance metric in the mathematical sense (i.e., satisfying symmetry and the triangle inequality) but to a general measure of dissimilarity between two distributions. Our goal is not to provide a comprehensive review of statistical distances, as there are already a number of excellent resources for that purpose, especially in specific domains of application (Borji, 2019; Xu et al., 2018; Lopez-Paz Oquab, 2016; Lueckmann et al., 2021). We refer readers to those works for a deeper dive into mathematical properties and empirical comparisons (Cox et al., 1984; Gibbs Su, 2002; Basseville, 2013; Cai Lim, 2022; Muandet et al., 2017; Theis et al., 2016; Betzalel et al., 2022). In this guide , we instead focus on four commonly applied sample-based distances in the machine learning literature for evaluating eventually high-dimensional generative models. They represent different methodologies for defining statistical distance: Using low-dimensional projections (Sliced-Wasserstein; SW), obtaining a distance using classifiers (Classifier Two-Sample Tests; C2ST), using embeddings through kernels (Maximum Mean Discrepancy; MMD) or neural networks (Fréchet Inception Distance; FID). We aim to provide an intuition for how and when to apply these distances, and to build a solid foundation for navigating the extensive literature on statistical distances.~~ Here, we present a guide that aims to serve as an accessible entry point to understanding commonly used sample-based statistical distances. Towards this goal, we provide explanations, comparisons, and example applications of four commonly applied distances: Sliced-Wasserstein (SW), Classifier Two-Sample Tests

(C2ST), Maximum Mean Discrepancy (MMD), and Fréchet Inception Distance (FID). With these resources, we aim to empower researchers to choose, implement, and evaluate the usage and outcomes of statistical distances for generative models in science. Note that, with *distance*, we do not necessarily refer to a distance metric in the mathematical sense (i.e., satisfying symmetry and the triangle inequality) but to a general measure of dissimilarity between two distributions (however, as we will point out, some of the distances we consider are in fact metrics).

~~Towards this goal, we provide for~~ The general and didactic nature of this guide means it can neither be comprehensive nor provide a clear-cut answer as to which distance is 'best', as there are numerous choices and the 'ideal' one strongly depends on the domain of application. On that front, previous articles have provided useful and extensive reviews for specific use cases. For example, Theis et al. (2016) discusses commonly used criteria for evaluating generative models of images, Borji (2019); Xu et al. (2018); Yang et al. (2023) compare metrics specific to evaluating GANs (including e.g., specialized variants of the FID), Basseville (2013) provides an extensive overview of previous works on divergences, and Gibbs & Su (2002) analyses the theoretical relationships among 'classic' distances (including e.g., Wasserstein and KL-divergence). However, by going through four different classes of sample-based distances in detail and systematically comparing them on synthetic and real-world applications we aim to provide a solid foundation for navigating the extensive literature on statistical distances, and to enable readers to reason about other related distances not covered here.

The outline of this guide is as follows: First, we provide an intuitive and graphical explanation for each of the four distances ~~an intuitive and graphical explanation~~ (Section 2). We then perform a systematic evaluation of their robustness as a function of dataset size, data dimensionality, and other factors, such as data multimodality (Section 3). Finally, in Section 4, we demonstrate how these distances can be applied to compare generative models in different scientific domains: We evaluate low dimensional models of decision making in behavioral neuroscience and generative models of medical X-ray images. We show the importance of using multiple complementary distances, as distinct distances can give different results when comparing the same sets of samples. ~~By presenting these resources, we aim to empower researchers to choose, implement, and evaluate the use and outcomes of statistical distances for generative models in science.~~

2 Sample-based statistical distances

In this section, we provide an overview of four classes of sample-based statistical distances commonly used in machine learning literature. Each class takes a different approach to overcoming the challenges inherent in comparing samples from high-dimensional and complex distributions. Throughout the section, we assume that we want to evaluate the distance between two datasets of samples, denoted as $\{x_1, x_2, \ldots, x_n\} \sim p_1(x)$ and $\{y_1, y_2, \ldots, y_m\} \sim p_2(y)$, where $p_1(x)$ and $p_2(y)$ are two probability distributions. These can be either two generative models, or a generative model and the underlying distribution of the observed data.

2.1 Slicing-based: Sliced-Wasserstein (SW) distance

Computing distance between distributions suffers from the *curse of dimensionality*, where the computational cost of computing the distance increases very rapidly as the dimensionality of the data increases. This problem is especially restricting when the distance is used as part of a loss function in optimization problems, since in this case it needs to be evaluated many times. This has prompted the notion of "sliced" distances, which have become increasingly popular in recent years (Kolouri et al., 2019; Nadjahi et al., 2020; Goldfeld & Greenewald, 2021). The main idea behind slicing is that for many existing statistical distances we can efficiently evaluate the distance in low-dimensional spaces, especially in one dimension. Therefore, the "slices" are typically one-dimensional lines through the data space (Fig. 2a). All data points from each distribution are projected onto this line by finding their nearest point on the line, giving a one-dimensional distribution of projected data points (Fig. 2b). The distance measure of interest between the resulting one-dimensional distributions can then be computed efficiently. However, computing the random projection could lead to an unreliable measure of distance: Distinct distributions can produce the same one-dimensional projections. Therefore, we repeat the slicing process for many different slices and average the resulting distances. More formally, we compute the expected distance in one dimension between the projections of the respective

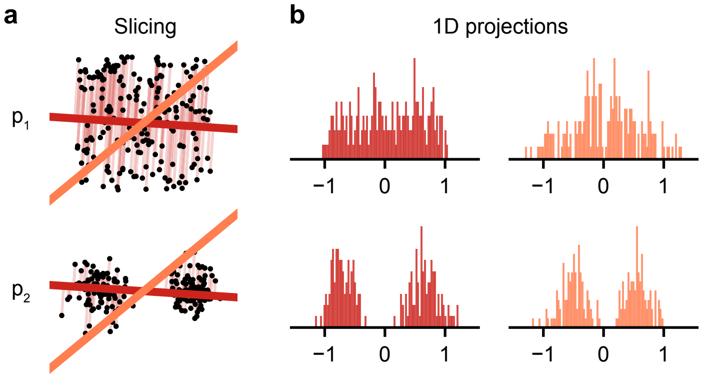

Figure 2: **Schematic for the Sliced-Wasserstein distance.** **(a)** Samples from two two-dimensional distributions along with example slices. The "slicing" is done by sampling random directions from the unit sphere and projecting the samples from the higher-dimensional distribution onto that direction. **(b)** One-dimensional projections of the two distributions corresponding to the two random slices in (a). For each pair of projections, the empirical Wasserstein distance is computed. Unlike in higher dimensions, this can be done efficiently for one-dimensional distributions.

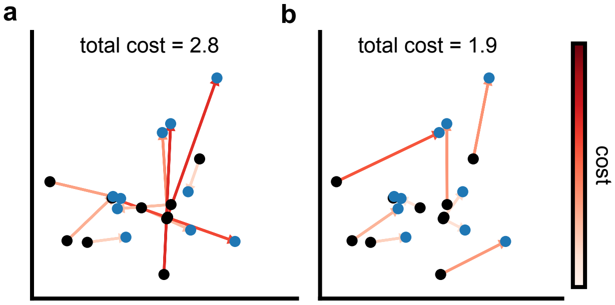

Figure 3: **Computing Wasserstein distance.** Two transport maps mapping the samples from a distribution p_1 (black) to samples from another distribution p_2 (blue), shown by arrows. The color of the arrow corresponds to the cost (Euclidean distance) between x_i and y_i. **(a)** Randomly chosen transport map. **(b)** The optimal transport map, giving the smallest total cost. The total cost for the optimal map in (b) is the Wasserstein distance between these two sets of samples.

distributions onto (uniformly) random directions on the unit sphere. As long as the distance of choice is a valid metric in one dimension, the sliced distance defined in this way is guaranteed to be a valid metric as well (Nadjahi et al., 2020, Proposition 1 (iii)). The most popular example of a sliced distance metric is the Sliced-Wasserstein (SW) distance (Fig. 2). However, we note that slicing has also been done for other distance measures, such as MMD with a specific choice of kernel ~~(?)~~ (Hertrich et al., 2024) and mutual information (Goldfeld & Greenewald, 2021). We provide a formal definition of the Wasserstein distance below, and of the Sliced-Wasserstein distance in Appendix A.3.

Definition of Wasserstein ~~or earth mover distance~~Distance Wasserstein distance is typically defined between two measures μ, ν. This definition is given in Appendix A.3, and here we provide the definition in the common case that μ and ν admit probability density functions p_1 and p_2 respectively. Let $M \subseteq \mathbb{R}^d$, and $||\cdot||_q$ be the q-norm in $\mathbb{R}^d$. Then the Wasserstein-q norm can be written as

$$W_q(p_1, p_2) = \inf_{\gamma \sim \Gamma(p_1, p_2)} \left(\mathbb{E}_{x_1, x_2 \sim \gamma} \, ||x_1 - x_2||_q^q \right)^{\frac{1}{q}}, \tag{1}$$

where $\Gamma(p_1, p_2)$ is the set of all couplings, that is all possible "transportation plans", between p_1 and p_2. $\gamma \in \Gamma(p_1, p_2)$ is a joint distribution over (x_1, x_2) with respective marginals p_1 and p_2 over x_1 and x_2.

Sample-Based Wasserstein Distance In practice, Eq. (1) is analytically solvable for only a few distributions. Therefore, Wasserstein distance is typically estimated from finite samples from p_1 and p_2. However, sample-based estimates of the Wasserstein distance are biased, and the convergence to the true Wasserstein distance is exponentially slower as the dimensionality of the distribution increases (Fournier & Guillin, 2015; Papp & Sherlock, 2022)

Intuitively, if two given probability distributions are thought of as two piles of dirt, the Wasserstein distance measures the (minimal) cost of "transporting" one pile of dirt to another (Panaretos & Zemel, 2019). ~~The formal definition of the Wasserstein distance for continuous distributions, derived from optimal transport, is described in Section A.3. Here, we provide a more intuitive definition given a fixed set of samples from two distributions.~~ Suppose we have samples $\{x_1, \ldots, x_N\} \subset \mathbb{R}^d$ sampled from a distribution p_1 and $\{y_1, ..., y_N\} \subset \mathbb{R}^d$ sampled from another distribution p_2. Given any distance metric between two vectors in $\mathbb{R}^d$, $D(\cdot, \cdot)$, we can construct the *cost matrix* C, as the matrix of pairwise distances between the samples x_i and y_j:

$$C = \begin{bmatrix} D(x_1, y_1) & \ldots & D(x_1, y_N) \\ \vdots & \ddots & \vdots \\ D(x_N, y_1) & \ldots & D(x_N, y_N) \end{bmatrix} \tag{2}$$

Recalling the earth-mover distance analogy, we want to map each x_i to exactly one y_j, in such a way that the cost of doing so is minimized. The minimum transport map is then defining the Wasserstein distance (for the metric D) between the two empirical distributions. Throughout this work, we use the commonly used Euclidean metric, L^2, leading to the Wasserstein-2 and Sliced Wasserstein-2 distances. More precisely, we define a "transport map" to be a permutation matrix, $\pi \in \{0, 1\}^{N \times N}$, which is a matrix with exactly one nonzero entry in each row. The entry $\pi_{ij} = 1$ means that we transport the point x_i to the point y_j. Then finding the transport map that minimizes the overall cost can be stated as

$$\pi^* = \min_{\pi} \sum_{ij} \pi_{ij} C_{ij}. \tag{3}$$

A randomly chosen transport map for small datasets in $\mathbb{R}^2$ is shown in Fig. 3a. Fortunately, the optimal solution to Eq. (3) can be solved exactly using the *Hungarian method* (Kuhn, 1955), leading to the assignment shown in Fig. 3b.

Slicing Wasserstein brings efficiency Solving the optimal transport problem (Eq. (3)) with the Hungarian method has a time complexity of $O(N^3)$ in the number of samples N (although faster ~~approximations exists~~($O(N^2 \log N)$ approximations exist, see Peyré et al. 2017). However, in the special case where the data is one-dimensional, the Wasserstein distance can be calculated by sorting the two datasets, obtaining the *order statistics* $\{x_{(1)}, .., x_{(N)}\}$ and $\{y_{(1)}, ..., y_{(N)}\}$ and computing the sum of the distances $\sum_i D(x_{(i)}, y_{(i)})$. This has a time complexity of $O(N \log(N))$. Thus, slicing the Wasserstein distance with one-dimensional projections becomes very efficient. While the value of the SW distance does not converge to the Wasserstein distance, even in the case of an infinite number of data samples and slices, the SW distance is a metric (in the mathematical sense) as long as D is a metric on $\mathbb{R}^d$ and it acts as a lower bound to the Wasserstein distance (Nadjahi, 2021).

The Wasserstein distance and its sliced variant have several attractive properties: they can be computed differentiably; their computations do not rely heavily on choices of hyperparameters; and the sliced variant is very fast to compute. ~~However, a disadvantage of the Wasserstein distance is its transparency: The numerical value of the Wasserstein distance has no intuitive interpretation due to its definition in terms of optimal transport maps~~However, a disadvantage of the Wasserstein distance is that its numerical value has no intuitive interpretation. Therefore, it is typically used to compare whether some distances are larger or smaller than others, instead of making qualitative statements about two distributions. Additionally, per definition, the sliced variant is insensitive to differences within orthogonal subspaces of the slices. To still capture differences in all dimensions, naively, one would have to increase the number of slices (in the worst case) exponentially with the dimension, diminishing computational efficiency. However, other approaches exist to reduce this problem for nonlinear (Kolouri et al., 2019) or other specific (Deshpande et al., 2019;

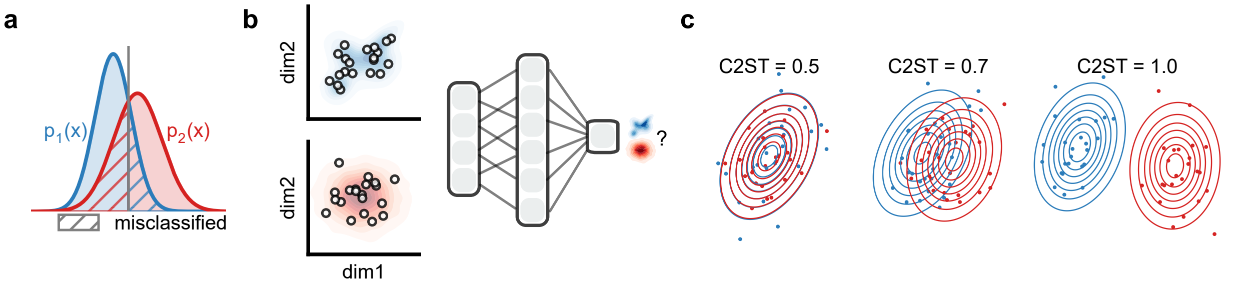

Figure 4: **Classifier Two-Sample Test (C2ST). (a)** The C2ST classifier problem: identifying the source distribution of a given sample. The optimal classifier predicts the higher-density distribution at every observed sample value, resulting in a majority of samples being correctly classified. **(b)** When probability densities of the distributions are not known, the optimal classifier is approximated by training a classifier, e.g., a neural network, to discriminate samples from the two distributions. **(c)** C2ST values vary from 0.5 when distributions exactly overlap (left) to 1.0 when distributions are completely separable (right).

2018) slices. Furthermore, slicing may also be relaxed to other kinds of data specific projections, such as Fourier features for stationary time series or locally connected projections for images (Du et al., 2023; Cazelles et al., 2020).

2.2 Classifier-based: Classifier Two-Sample Test (C2ST)

The Classifier Two-Sample Test (C2ST) uses a classifier that discriminates between samples from two distributions (Fig. 4a) (Lopez-Paz & Oquab, 2016; Friedman, 2003). The distance between the distributions can then be quantified with various measures of classifier performance. For example, one would train a classifier $c(x)$ to distinguish samples from the generative model and the data, and then evaluate the C2ST as $\frac{1}{2}[\mathbb{E}_{p(x)}[\mathbb{1}(c(x) = 0)] + \mathbb{E}_{q(x)}[\mathbb{1}(c(x) = 1)]]$. The classification accuracy provides a particularly intuitive and interpretable measure of the similarity of the distributions. If the classification accuracy is 0.5, i.e., the classifier is at chance level, the distributions are indistinguishable to the classifier (Fig. 4c, left), while higher accuracy indicate differences in the distributions (Fig. 4c, middle). If the C2ST is 1.0, the two distributions have no (or very little) overlap in their supports (Fig. 4c, right). Given two distributions, the C2ST has a 'true' (optimal) value, which is the maximum classification accuracy attainable by any classifier (Fig. 4a). This optimal value can be computed if both distributions allow evaluating their densities, but this is not usually possible if only data samples are available. In that case, one aims to train a classifier, such as a neural network (Fig. 4b), that is as close to the optimal classifier as possible.

One of the main benefits of the C2ST is that its value is highly interpretable (the accuracy of the classifier). C2ST can also be used to test the statistical significance of the difference between two sets of samples. Unlike other measures, however, C2ST can be expensive to compute because it requires training a classifier and using the classification accuracy as a differentiable training objective is not straightforward (see Section A.6.1). Furthermore, the value is dependent on the capacity of the classifier, and hence on many hyperparameters such as classifier architecture or training procedure. This dependence on a trained classifier ensures that C2ST estimates are biased, though the variety of possible classifier architectures means theoretical guarantees such as sample complexity are difficult to determine. In our experiments, we used a scikit-learn ~~Multi-Layer-Perceptron~~ Multi-Layer Perceptron classifier, combined with a five-fold cross-validation routine to estimate the accuracy returned (Pedregosa et al., 2011).

Common failure modes As mentioned above, for any realistic scenario, the C2ST is computed by training a classifier. The resulting C2ST will only be a good measure of distance between real and generated data if the classifier is close or equal to the optimal classifier. To demonstrate the behavior of the C2ST if this is not the case, we fitted a Gaussian distribution to data that was sampled from a Mixture of Gaussians

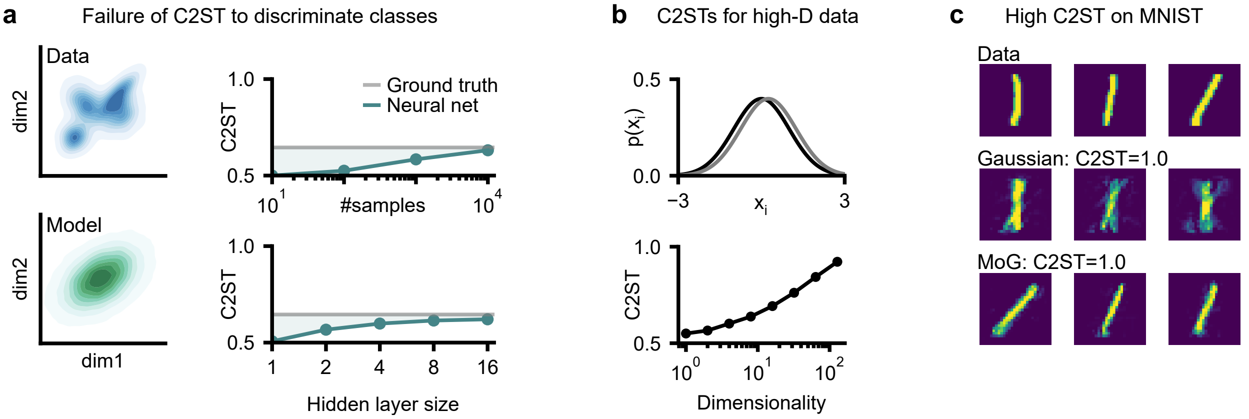

Figure 5: **Failure modes and behavior of C2ST. (a)** Data (top left) and Gaussian maximum-likelihood estimate (bottom left). C2ST wrongly returns 0.5 (no difference between the densities) if too few samples are used (top right) or the neural network is poorly chosen (bottom right). **(b)** For high-dimensional densities, despite the marginals between data (black) and model (gray) seeming well-aligned, small differences (here a mean shift of 0.25 std. in every dimension) allow the classifier to more easily distinguish the distributions as dimensionality increases, yielding correct but surprisingly high C2ST. **(c)** On MNIST, the C2ST between data (top) and a Gaussian generative model (middle) as well as of a Mixture of Gaussians (MoG, bottom) is 1.0, although the MoG is perceptually more aligned with the data.

(Fig. 5a, left). The optimal C2ST between these two distributions is 0.65 (which can be computed because Gaussians and Mixtures of Gaussians allow evaluating densities). If the C2ST is estimated with a neural network, however, we observe that this C2ST can be systematically underestimated: for example, when only few samples from the data and generative model are available, the neural network predicts a C2ST of 0.5 (Fig. 5a, top right) – in other words, it predicts that the generative model and the data follow the same distribution. Similarly, if the neural network is not expressive enough, e.g., with too few hidden units, the classifier will return a low C2ST, around 0.5 (Fig. 5a, bottom right). These issues can make the C2ST easy to misuse: In many cases, reporting a low C2ST is desirable for generative models since it indicates that the model perfectly matches data, but one can achieve a low C2ST simply by not investing sufficient time into obtaining a strong classifier.

C2ST can remain very high even for seemingly good generative models We previously argued that the C2ST is an interpretable measure – while this is generally true, the C2ST can sometimes be surprisingly high even if the generative model seems well aligned with the data. For example, when the generative model aligns very well with the data for every marginal, the C2ST can still be high if the data is high-dimensional (Fig. 5b). Because of this, it can be difficult to achieve low C2ST values on high-dimensional data. To further demonstrate this, we fitted a Gaussian distribution and a mixture of 20 Gaussian distributions to the ‘ones’ of the MNIST dataset. Although the Mixture of Gaussians (Fig. 5c, bottom row) looks better than a single Gaussian (Fig. 5c, middle row), both densities have a C2ST of 1.0 to the data (obtained with a ResNet on ≈4k held-out test datapoints).

Other C2ST variants While we focus on a standard C2ST definition by using classification accuracy as the C2ST distance (Lopez-Paz & Oquab, 2016), any other performance metric for binary classification could be used (Raschka, 2014). Kim et al. (2019) even ~~argue~~ argues that classic accuracy is sub-optimal due to the “binarization” of the class probabilities and proceeds to instead use the mean squared error between the predicted and ‘target’ value of 0.5. Other approaches instead construct a likelihood ratio statistic (Pandeva et al., 2022). Additionally, instead of using the estimated class probabilities, Cheng & Cloninger (2022) consider using the average difference in logits (i.e., activations in the last hidden layer).~~However, classification accuracy is still the most commonly used variant of C2ST.~~

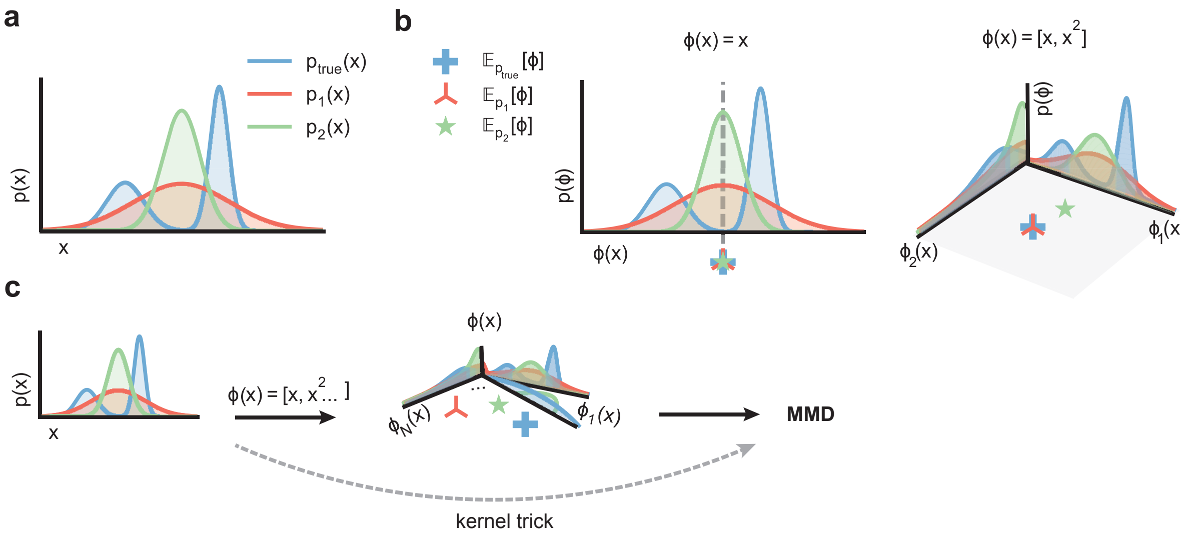

Figure 6: **Maximum mean discrepancy (MMD).** **(a)** Two example distributions $p_1(x)$, $p_2(x)$ and observed data $p_{true}(x)$ that we want to compare. **(b)** MMD can be defined as the difference between the expectations of some embedding function $\phi(x)$. If we take the identity as embedding ($\phi^{(1)}(x) = x$; left), we end up computing the differences between the means of the distributions, which are all equal for the three distributions. If we add a quadratic feature ($\phi(x)^{(2)} = [x, x^2]^\intercal$; right), we can distinguish distributions with different variances. Note that we still have $\mathsf{MMD}^2[\phi^{(2)}, p_2, p_{true}] = 0$, despite p_2 being different from p_{true} **(c)** Using the *kernel trick* we can avoid computing the embeddings all together but use implicit embeddings that capture all relevant features of the distributions.

We note that the learned classifier in C2ST can be applied to estimations of a density ratio $\frac{p(x)}{q(x)}$, that is referred to as the likelihood ratio trick (Hastie et al., 2001; Sugiyama et al., 2012). Density ratio estimation has attracted a great deal of attention in the statistics and machine learning communities since it can be employed for estimating divergences between two distributions, such as the Kullback–Leibler divergence (Titsias & Ruiz, 2019; **?**) and Pearson divergence (Srivastava et al., 2020). Finally, we note that the classifier in C2ST is used as discriminator in Generative Adversarial Networks (GAN) Goodfellow et al. (2014).

2.3 Kernel-based: maximum mean discrepancy (MMD)

MMD is a popular distance metric that is applicable to a variety of data domains, including high-dimensional continuous data spaces, strings of text as well as graphs (Borgwardt et al., 2006; Gretton et al., 2012a; Muandet et al., 2017). It has been used to evaluate generative models (Sutherland et al., 2021; Borji, 2019; Lueckmann et al., 2021) and also has the ability to indicate *where* the model and the true distribution differ (Lloyd & Ghahramani, 2015). The distance provided by MMD can straightforwardly be used to test whether the difference between two sets of high-dimensional samples is statistically significant (Gretton et al., 2012a).

To assess whether two ~~set~~ sets of samples are drawn from the same distribution, MMD makes use of a kernel function to (implicitly) embed the samples via an embedding function ϕ, also called a feature map. If we choose the right kernel, we can end up embedding our samples in a space where the properties of the underlying distributions are easily compared. We will motivate the use of the kernel in MMD by illustrating different explicit embeddings before introducing the implicit embedding via a kernel k. Note that this explanation is inspired by Sutherland (2019).

In a first step, we can define MMD as the difference between the means of the embedding of two distributions p_1 and p_2:

$$\mathsf{MMD}^2[\phi, p_1, p_2] = \|\mathbb{E}_{p_1(x)}[\phi(x)] - \mathbb{E}_{p_2(y)}[\phi(y)]\|^2,$$

for any embedding function ϕ.

If we want to compare samples of real numbers from two distributions p_1 and p_2 (Fig. 6a), we can think about different embedding functions ϕ to compare these. The simplest possible function $\phi^{(1)} : \mathbb{R} \to \mathbb{R}$ is the identity mapping $\phi^{(1)}(x) = x$ (Fig 6b, left). However, in this case, the MMD will simply be the absolute difference between the means (first moments) of the distributions (for details, see Section A.2):

$$\mathsf{MMD}[\phi^{(1)}, p_1, p_2] = |\mu_{p_1} - \mu_{p_2}|.$$

As both models and the true distribution in Fig. 6 have the same mean, this does not yet ~~let us~~ allow us to discriminate between them. If we now expand our embedding with a quadratic term, $\phi^{(2)} : \mathbb{R} \to \mathbb{R}^2$ as $\phi^{(2)}(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$ (Fig 6b, right), the MMD yields (for details, see Section A.2)

$$\mathsf{MMD}^2[\phi^{(2)}, p_1, p_2] = (\mu_{p_1} - \mu_{p_2})^2 + (\mu_{p_1}^2 + \sigma_{p_1}^2 - \mu_{p_2}^2 - \sigma_{p_2}^2)^2.$$

In this case, we can also distinguish distributions with different variances (second moments). This allows us to differentiate between two out of three distributions (Fig. 6). If we want to distinguish between all three distributions, we could keep adding additional features to ϕ to capture higher ~~and higher~~ moments. However, this seems like it could get infeasible – if we want to make sure two probability distributions are exactly equal, i.e., have exactly the same moments, we would need to add infinitely many moments. Luckily, there is a trick we can exploit. First, we can rewrite MMD in terms of inner products of features (denoted with $\langle \cdot, \cdot \rangle$; for details, see Section A.2) as

$$\mathsf{MMD}^2[\phi, p_1, p_2] = \mathbb{E}_{p_1(x), p_1'(x')}[\langle \phi(x), \phi(x') \rangle] + \mathbb{E}_{p_2(y), p_2'(y')}[\langle \phi(y), \phi(y') \rangle] - 2\mathbb{E}_{p_1(x), p_2(y)}[\langle \phi(x), \phi(y) \rangle]$$

We can now rewrite the inner product $\langle \phi(x), \phi(x') \rangle$ in terms of a kernel function k: $\langle \phi(x), \phi(x') \rangle = k(x, x')$. Thus, if we can find a kernel for our feature map, we can avoid explicitly computing the features altogether but instead, we directly compute

$$\mathsf{MMD}^2[k, p_1, p_2] = \mathbb{E}_{p_1(x), p_1'(x')}[k(x, x')] + \mathbb{E}_{p_2(y), p_2'(y')}[k(y, y')] - 2\mathbb{E}_{p_1(x), p_2(y)}[k(x, y)].$$

Evaluating the kernel function instead of explicitly calculating the features is often called the *kernel trick* (Fig. 6c). If we can define a kernel whose corresponding embedding captures all, potentially infinitely many moments, we would have an MMD that is zero only if two distributions are exactly equal and the MMD becomes a *metric*. These kernels are called *characteristic* (Section A.2, Gretton et al. 2012a), and include the commonly used Gaussian kernel: $k_G(x, x') = \exp(-\frac{\|x-x'\|^2}{2\sigma^2})$~~(see, e. g., Sriperumbudur et al. (2009) for other characteristic kernel choices).~~. A number of other kernels can also be used (see e.g., Sriperumbudur et al. (2009)). For instance, using a Euclidean distance-based kernel, MMD can be shown to be equivalent to the standard energy distance (Székely & Rizzo, 2013). In fact, a wider equivalence between MMD and the generalized energy distance has been established, using certain distance-induced kernels (Sejdinovic et al., 2013).

MMD in practice Typically, the kernel version of MMD is used, which is straightforwardly estimated with its empirical~~(unbiased)~~, *unbiased*, estimate:

$$\mathsf{MMD}^2 = \frac{1}{m(m-1)} \sum_i^m \sum_{j \neq i} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_i^n \sum_{j \neq i} k(y_i, y_j) - \frac{2}{mn} \sum_{i,j} k(x_i, y_j).$$

MMD in this form can be applied to many forms of data, as long as we can define a kernel, which can include graphs (Vishwanathan et al., 2010; Gärtner, 2003) or strings of text (Lodhi et al., 2002), in addition to vectors and matrices.

When we estimate the MMD with a finite number of samples, the selection of the right kernel and its parameters becomes crucial. For example, when using a Gaussian kernel, one has to choose the bandwidth σ. The MMD approaches zero if we take σ to be close to zero (then $k_G(x, x') = 1 \text{ if } x = x' \text{ else } k_G(x, x') \to 0$)

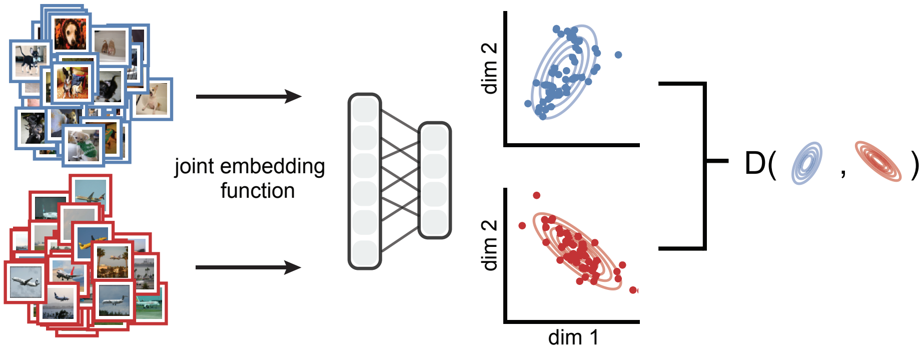

Figure 7: **Network-based metrics.** Instead of directly computing distances in data space, complex data e.g., natural images of dogs sampled from $p_1(x)$ and aircraft sampled from $p_2(y)$, are jointly embedded into a vector space. The embedding function can, for example, be a deep neural network. The resulting distributions in feature space are then compared by a classical measure of choice D.

or if σ is large (then $k_G(x, x') \to 1 \; \forall \; x, x'$) (Gretton et al., 2012a). A common heuristic to remedy this parameter choice is picking the bandwidth based on the scale of the data. The *median heuristic* ~~set~~ sets the bandwidth to the median distance between points in the aggregate sample (Gretton et al., 2012a). Another common approach is based on cross-validation, or data splitting (Gretton et al., 2012a;b; Jitkrittum et al., 2016; Sutherland et al., 2021): The dataset is divided, with a hold-out set used for kernel selection, and the other part used for evaluating MMD. While the data splitting method does not involve any heuristic, it can lead to errors in MMD since it reduces the number of data points available for estimating the MMD. Recent work attempts to choose hyperparameters without employing data splitting or any heuristic (Biggs et al., 2023; Schrab et al., 2023; Kübler et al., 2022b;a).

While we ~~aim in general~~ often aim for a kernel that captures the (dis)similarity between the data points well, such a kernel can be ~~domain specific~~ domain-specific or specifically designed for downstream analysis tasks. The similarity between two strings (e.g., DNA sequences ~~,~~ or text) can~~for instance~~, for instance, be estimated by looking at the frequency of small subsequences (Leslie et al., 2001; Lodhi et al., 2002).~~It is furthermore~~ Furthermore, it is possible to aggregate simpler kernels into a more expressive one (Gretton et al., 2012b), or to use a deep kernel (i.e., based on neural networks) that can exploit features of particular data ~~modality~~ modalities such as images (Liu et al., 2020; Gao et al., 2021).

2.4 Network-based: Embedding-space measures

Distribution comparisons on structured data spaces, such as ~~the~~ a set of natural images, present unique challenges. Such data is usually high-dimensional (high-resolution images) and contains localized correlations. Furthermore, images of different object classes (such as airplanes and dogs) share low-level features in the form of edges and textural details but differ in semantic meaning. Similar challenges occur for time-series data, natural language text, and other complex data type (Smith & Smith, 2020; Jeha et al., 2021).

In this section, we ~~take~~ rely on the example of natural images, but the presented framework generalizes to other data types. Naive distances would operate on a per-pixel basis, leading to scenarios where, for example, white dogs and black dogs are considered vastly different despite both being categorized as dogs. As we would like to have a distance measure that operates based on details relevant to the comparison, we can leverage neural networks trained on a large image dataset that captures features ranging from low-level to high-level semantic details: While earlier layers in a convolutional neural network focus on edge detection, color comparison, and texture detection, later layers learn to detect high-level features, such as a dog's nose or the wing of an airplane, which thought to be relevant for a meaningful comparison. Embedding-based distances use these activations of neural network layers as an embedding to compare the image distributions. The most popular distance in this class is the Fréchet Inception Distance (FID) (Heusel et al., 2017), used to evaluate generative models for images. The FID uses a convolutional neural ~~net~~networks's embeddings (specifically InceptionV3 (Szegedy et al., 2015a)) to extract ~~the~~ relevant features, applies a Gaussian approximation in the embedding space, and computes the Wasserstein distance on this approximation.

~~An~~ A FID-like measure, in essence, requires a suitable *embedding network* $f : \mathcal{X} \to \mathbb{R}^d$, where f transforms the data from the original high-dimensional space $\mathcal{X}$ into a lower-dimensional, feature-rich representation in $\mathbb{R}^d$ (Fig. 7). Once the data samples are mapped into this reduced space through the embedding network, the two sets of embedded samples can be compared using the appropriate distance. When evaluating generative models for natural images, it is common to approximate the *embedded* distributions with Gaussian distributions by estimating their respective mean μ and covariances Σ. Under this Gaussian approximation, the squared Wasserstein distance (also known as the Fréchet distance) can be analytically computed as

$$\underline{W^2((\mu_1, \Sigma_1), (\mu_2, \Sigma_2)) = \|\mu_1 - \mu_2\|^2 + \mathrm{Tr}\left(\Sigma_1 + \Sigma_2 - 2\left(\Sigma_1 \Sigma_2\right)^{\frac{1}{2}}\right).}$$

$$W^2((\mu_1, \Sigma_1), (\mu_2, \Sigma_2)) = \|\mu_1 - \mu_2\|^2 + \mathrm{Tr}\left(\Sigma_1 + \Sigma_2 - 2\left(\Sigma_1 \Sigma_2\right)^{\frac{1}{2}}\right). \tag{4}$$

~~In principle, any appropriate metric can be used in place of the Fréchet distance. Jayasumana et al. (2023) show that the MMD (Section 2.3) can be better suited as a metric in the embedding space, as it perceptually matches human judgement on assessing image quality and coverage in generative models.~~ In principle, any appropriate metric can be used in place of the Fréchet distance. For instance, Jayasumana et al. (2023); Bińkowski et al. (2021); Xu et al. (2018) use MMD as a metric in the embedding space, and the MMD-based Inception Distance is often referred to as Kernel Inception Distance (KID). KID is known to have some advantages over FID: unlike FID, KID has a simple, unbiased estimator and does not assume any parametric forms for the distributions. Moreover, KID requires a smaller sample for reliable estimation compared to FID. Since KID involves MMD, we must carefully select the proper kernel and its hyperparameter when applying it. A related and commonly used quality measure for images is the Inception Score (Salimans et al., 2016). In contrast to the FID, this measure uses the average InceptionV3 predicted class probabilities and compares them with the true marginal class distribution. Note that while both this score and the FID can agree with traditional distances (e.g., certain divergences), they might evaluate models differently (Betzalel et al., 2022); see Barratt & Sharma (2018) for further limitations of the Inception Score.

Limitations One of the biggest limitations is the requirement of a suitable embedding ~~net~~network. Newer and more robust networks, such as the image network of the CLIP (Radford et al., 2021) vision-language model, provide better and semantically more consistent embeddings (Betzalel et al., 2022) than the InceptionV3 network. However, as the embedding network is generally non-injective, identical distributions in the embedding space may not necessarily translate to identical distributions in the original space. Previous research has demonstrated the FID's sensitivity to preprocessing such as image resizing and compression (Parmar et al., 2022). Additionally, FID estimates are biased for finite sample sizes, making comparisons unreliable due to dependency on the generative model. However, methods to obtain a more unbiased estimate have been proposed (FID_∞; Chong & Forsyth 2020; Betzalel et al. 2022).

3 Comparison and scalability

When evaluating (or training) generative models, it is important to understand that different statistical distances ~~penalize~~ pay attention to different features of the generated samples. They might~~for instance~~, for instance, weigh differently how important it is to have large sample variability versus how well the modes of the true distribution are captured~~(as can be illustrated~~. We illustrated the trade-off of the first two properties by optimizing a mis-specified model using the different distances, see Fig. S2 and Theis et al. 2016~~). Thus, using different statistical distances to evaluate the quality of generative models can lead to different conclusions.~~.

These differences can be even more pronounced in applications where we only have a limited amount of data points, e.g., identifying rare cell types (Marouf et al., 2020), or where we have very high-dimensional data,

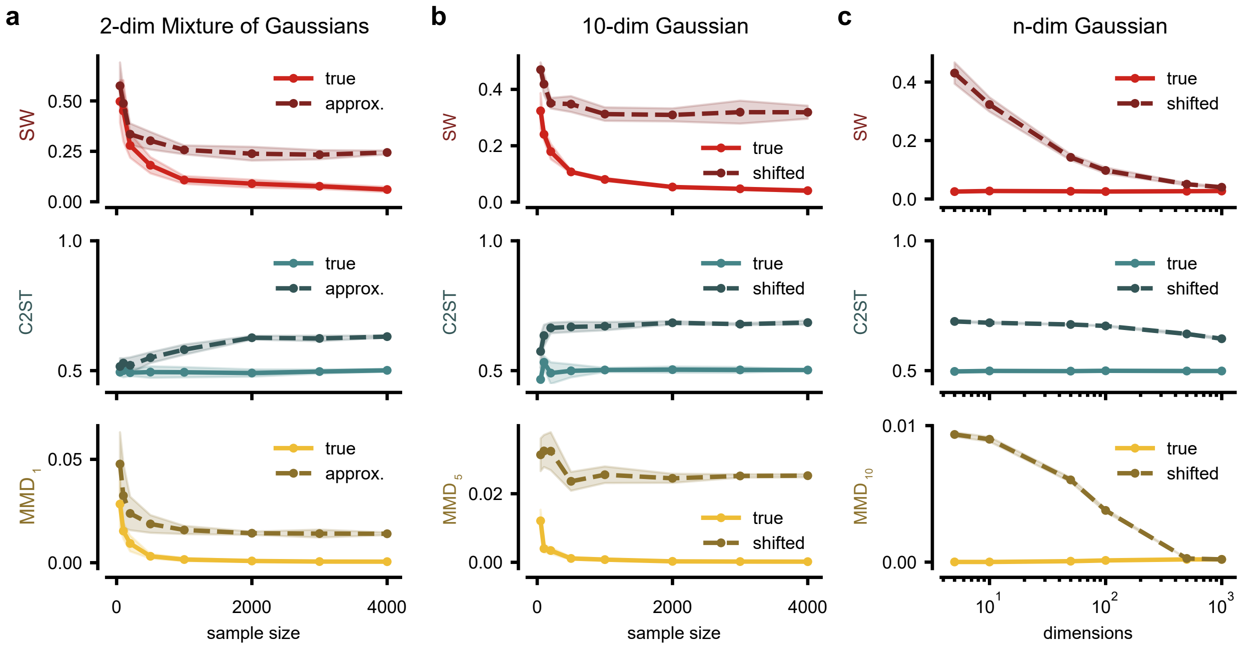

Figure 8: **Scalability of different statistical distances with sample size and dimensionality. (a,b)** Comparison of sample sets with varying sample size (between 50 and 4k samples per set) of a 'true' distribution, either with a second dataset of the same distribution or with a sample set from an approximated/shifted distribution. We show the mean and standard deviation over five runs of randomly sampled data. Note that the subscript for MMD distances (bottom) denotes the bandwidth of the Gaussian kernel used for a given dataset and we report the squared distance for MMD. **(a)** Distances for the 2d-MoG example shown in Fig. 1 compared to samples from a ~~unimodal~~ uni-modal Gaussian approximation with the same mean and covariance. **(b)** Distances for a ten-dimensional standard normal distribution, for which *the first* dimension is shifted by one for the shifted dataset. **(c)** Distances based on 10k samples from a standard normal distribution with varying dimensions (between 5 and 1000). As in (b), *the first* dimension is shifted by one for the 'shifted' dataset. We show the mean and standard deviation over five runs of randomly sampled data. One MMD bandwidth was selected for all n-dimensional datasets.

e.g., neural population recordings in neuroscience (Stringer et al., 2019). In such cases, one needs to ensure that the distance measures can reliably distinguish different distributions for the given sample set size while remaining computationally tractable. In the following sections, we investigate the sensitivity of the three presented distances which do not rely on embeddings, when it comes to distinguishing data sets with varying numbers of samples (Section 3.1) and varying numbers of dimensions (Section 3.2). As the absolute values of the distance measures are often hard to interpret and different measures are on different scales, we examined the relative distances by comparing two or more models to the true data. We, therefore, applied the distances to compare samples from a 'true' distribution against itself (intra-dataset) and against samples from another distribution that is either an approximation or a shifted version of the true distribution (inter-dataset).

In our experiments, we focused on inter- and intra-dataset comparisons for the following three datasets: First, we compared the two-dimensional Mixture of Gaussians ("2d-MoG") dataset introduced in Fig. 1 with samples from a unimodal Gaussian approximation with the same mean and covariance as the true data samples. Second, we compared samples from a *ten-dimensional* standard normal distribution with a shifted normal distribution for which the first dimension is shifted by one ("10-dim Gaussian"). And last, we compared a standard normal distribution with *varying dimensionality* to shifted distributions for which we respectively shifted the first dimension for the inter-dataset comparison ("n-dim Gaussian"). We used default parameters for the SW and C2ST measures while adjusting the bandwidth parameter for the MMD measure for each of the three comparisons. Finally, as FID and other network-based distances do require an embedding network, we investigate the scaling properties specific to FID on the ImageNet dataset (in Section 3.3).

3.1 Varying number of samples

We explored the robustness of the distances to low sample sizes on the 2d-MoG and the 10-dim Gaussian dataset. We found that for the 2d-MoG dataset, all measures failed to reflect the dissimilarity of the distributions at the lowest sample size of 50 samples (Fig. 8a). However, C2ST's behavior differs from MMD and SW, with C2ST indicating that the distributions are similar (C2ST ≈ 0.5) for both intra- (*true*) and inter-dataset (*approx.*) comparisons while the other two distances indicate they are different (distance $\neq 0$). The malfunction of C2ST can be harder to detect in such cases, compared to the one of MMD and SW. While the latter is easily identified by the incorrect intra-dataset results, the malfunction of C2ST is hard to detect for unknown distributions. For all measures, computed values quickly stabilized by a sample size of 1000 and yielded the expected results of low intra-dataset differences and high inter-dataset differences.

For the 10-dim Gaussian dataset, we observed that all distances can identify samples from the same distribution as more similar than samples from different distributions (Fig. 8b); while all three distances struggle with low sample sizes (see also Supp. Fig. S4), given enough samples, they all become robust, with no measure being clearly superior to the others. For the 2d-MoG experiment, more samples are required to clearly detect the difference between the two distributions (Fig. 8a) as compared to the case where the mean in one dimension is shifted (Fig. 8b). Intuitively, the larger the differences in the distributions we ~~want to~~ compare, the fewer samples we need to detect these differences (see additional experiments Supp. Fig. S4).

3.2 Varying dimensionality

We further tested how the distances scale with the data dimension using n-dimensional standard normal distributions. In the first experiment, the shifted distribution differed only in the first dimension, which was mean-shifted by one. As the dimensionality increases, all distances reliably indicate no difference in intra-dataset comparisons , but C2ST is the only measure that consistently identifies the inter-dataset difference (Fig. 8c). Note that this dataset is different from Fig. 5b, where mean shifts were applied to every dimension. When we changed the structure of the data distribution (e.g., by changing the mean of all dimensions or their variances, see Supp. Fig. S5), we observed a similar picture with some particularities: While the SW distance with a fixed number of slices has difficulties if the disparity between the distribution is only in one dimension, its performance drastically improves for differences in all dimensions, which is expected from the random projections SW is performing. C2ST seems to robustly detect differences even in high dimensions in these modified datasets, though previous experiments showed that this measure can be oversensitive to small changes in high dimensions (Fig. 5b,c). Lastly, MMD is not robust across different dimensions for a Gaussian kernel with *a fixed bandwidth*. We could make the MMD robust across dimensions by using median heuristic (Section 2.3), where we would increase the chosen bandwidth such that it stays on the order of the euclidean distance between datapoints. Note that ~~MMD can in general~~in general, MMD can be highly sensitive to ~~its hyperparameters, and an appropriate value~~ kernel- and hyperparameter choice. In addition, an appropriate setting depends not only on the dimensionality of the data (Supp. Fig. S6~~and S7~~, S7, and S8), but also on the structure of the distributions (Supp. Fig. S5). SW distance, on the other hand, is robust to number of random projections used (Supp. Fig. S1 and A.4).

3.3 FID-like distance comparison on ImageNet

To ~~also~~ explore scaling properties of FID-like distances, we used images from the ImageNet dataset and embedded them with the InceptionV3 network (Deng et al., 2009; Szegedy et al., 2015b), following the implementation of Heusel et al. (2017). In addition to the 100~~,~~ 000 images in the ImageNet test dataset (1000 classes, 100 images per class), we generated high-quality synthetic samples using a state-of-the-art diffusion model as described by Dockhorn et al. (2022). We first produced 50~~,~~ 000 samples with the base unconditional version of this model. Using a conditional generative model, we additionally generated 100~~,~~ 000 class-conditional images (i.e., 100 per class), exactly matching the class distribution of the ~~test-set~~test set.

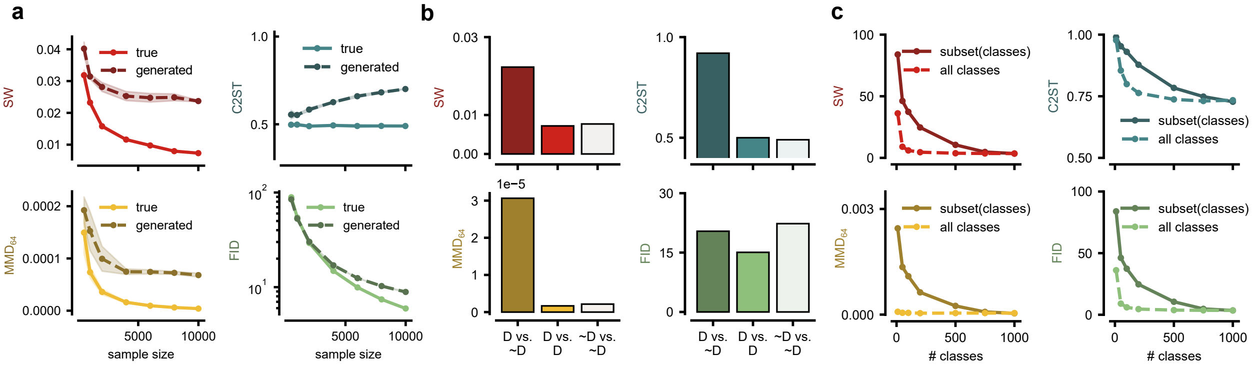

Figure 9: **Comparison of distances for ImageNet.** **(a)** A comparison between the ImageNet test set and samples generated by an unconditional diffusion model ~~,~~ with varying sample sizes. **(b)** Distance evaluation on dog classes (D) versus non-dog classes (~D), highlighting differences in image representation between these two categories of real data. **(c)** Distances between sets of randomly selected images vs. varying number of included classes of images (from 10 to 1,000) from the test set, using synthetic samples created by a conditional diffusion model.

All of these images were embedded using the pre-trained InceptionV3 network (Szegedy et al., 2015a), transforming the raw images into a 2048-dimensional feature space.

Calculating the FID involves computing the mean and covariance of the distributions in the embedding space and then calculating the squared Wasserstein distance analytically. However, we broadened our evaluation by applying additional distances to the distributions in the embedding space. While SW distance, C2ST, and FID effectively highlight the greater dissimilarity of synthetic samples to real images (i.e., the ImageNet test set) (Fig. 9a) even for low sample sizes, the distinctiveness of the FID becomes only apparent when analyzing more than around 2000 samples. Estimating the full covariance matrix of the ~~2048-dim~~ 2048-dimensional features with fewer samples leads to degeneracy and, thus, to numerical issues computing the square root. Common implementations~~hence~~ , hence, generally recommend using more than 2048 samples (Heusel et al., 2018a;b). As also shown by Jayasumana et al. (2023); Betzalel et al. (2022), the Gaussian assumption in the FID is violated and can lead to problematic behavior. In contrast, the other distances reliably estimate a larger inter-dataset distance in regimes with few samples.

In our subsequent analysis, we aimed to determine the effectiveness of various distances in discriminating between images from different classes by comparing the distances between different levels in the WordNet hierarchy (Miller, 1995). To this end, we focused on comparing images of dogs (D) with those of non-dog (~D) images (Fig. 9b). All investigated distances, except FID, were successful in identifying images from different classes as being more distinct than images from the same class. The FID comparison between data with multiple classes (~D vs. ~D) is higher than across dog classes and other classes (D vs. ~D). This shows that comparing two image classes can be problematic with FID, which is usually used to compare two distributions over many image classes (i.e., over natural images).

To examine the effects only including a subset of classes from the dataset (i.e., modes of the distribution), we employed a test set and a conditional synthetic dataset, each comprising ~~1,000~~ 1000 classes with 100 samples per class. Our analysis involved comparing the complete test set against synthetic datasets that included only subsets of classes (Fig. 9c). For comparison purposes and as a control measure, we also conducted a scenario where, instead of selectively excluding classes, we randomly removed an equivalent number of images from the dataset. This approach revealed that limiting the dataset to a small number of classes compromised the performance across all evaluated distances, in contrast to the outcomes observed when randomly excluding a subset of images. To achieve performance comparable to that observed with random removals, it was necessary to include at least 800 classes in the comparison. As the InceptionV3 network is, in essence, trained to classify ImageNet images (Szegedy et al., 2015a) (under certain regularization schemes), ~~hence~~ the extracted high-level features may also be very sensitive to class-dependent image features

Table 1: **Summary of practical and theoretical properties of metrics in terms of number of samples N and data dimensionality D.** Sample complexity here refers to the convergence rate of the sample-based estimate to the true value of the metric. *Bound based on Ghosal & Sen (2019); Nguyen & Ho (2024) for SWD and Gretton et al. (2012a) for MMD. †Best case scenario. In practice, the computational cost of training a neural network scales superlinearly with both sample size and data dimensionality. ‡ General case (see Sec. 2.3, 3.4 for details).※Cost of calculating the square root of the covariance matrix in Eq. 4.

	SW	C2ST	MMD	FID
Sample Complexity (N)	$\mathcal{O}(N^{-1/2})$*	N/A	$\mathcal{O}(N^{-1/2})$*	N/A
Computational Complexity (N)	$\mathcal{O}(N \log N)$	$\mathcal{O}(N)^{\dagger}$	$O(N^2)^{\ddagger}$	$\mathcal{O}(N^2)$
Computational Complexity (D)	$\mathcal{O}(D)$	$\mathcal{O}(D)^{\dagger}$	$\mathcal{O}(D)$	$\mathcal{O}(D^3)$※
Estimator unbiased?	no	no	yes	no

and not necessarily for general image quality. This behavior can be observed in Fig. 9c and was recently explored by Kynkäänniemi et al. (2022). By replacing InceptionV3 with other embedding networks (e.g.~~CLIP~~ , CLIP, which is trained to match images to ~~captions)~~corresponding text), this class sensitivity can be reduced (Kynkäänniemi et al., 2022).

We generated images using an additional consistency model (CS) for unconditional image generation (Song et al., 2023) to investigate how the metrics compare images created by different generative models. This model was trained on ImageNet 64x64 as GENIE (Dockhorn et al., 2022). Additionally, we included the models: BigGAN Brock et al. (2018), ablated diffusion model (ADM) Dhariwal & Nichol (2021), Glide (Nichol et al., 2021), Vector Quantized Diffusion Model (VQDM) Gu et al. (2022), Wukong Wukong (2022), Stable diffusion 1.5 (SD1.5) Rombach et al. (2022b) and Midjourney Midjourney (2022) (details in Appendix A.11). We evaluated the metrics (and multiple commonly used variants of KID and C2ST) for each model against the ImageNet test set (see Table S3).

While there is some agreement between the metrics regarding which model generates images closest to the ImageNet test set, there are also differences in the relative ordering across different metrics. As expected, the most recent unconditional models trained directly on ImagenNet 64x64 performed best in our evaluation (GENIE, CS), better than the two other unconditional generative models (BigGAN, ADM). The other models are text-to-image and thus only prompted to generate images from specific ImageNet classes (GLIDE, VQDM, Wukong, SD1.5, Midjourney). Interestingly, the prompted models performed better than older unconditional models (BigGAN, ADM) most of the time. Recall that all we evaluate is the similarity to the ImageNet test set; prompted versions might produce images from the correct classes but might contain differences in style or appearance compared to actual images in ImageNet. Despite the demonstrated class-sensitivity (Fig. 9c), the InceptionV3 embeddings are thus also sensitive to different "styles" of natural images. We want to note that in this case, being closer to ImageNet does not necessarily mean generating better images (based on human perception), but rather creating images that are more ImageNet-like.

3.4 ~~Computational~~ Sample and computational complexity

While we only considered the sensitivity and specificity of the different distances in the previous paragraphs, we want to highlight that they ~~differ also in their computational complexity~~also differ in their sample and computational complexities (Table 1). With respect to the number of samples N, MMD and FID have a complexity of $O(N^2)$~~(note~~ . Note that for MMD the computational cost can be reduced, ~~potentially at the cost of making approximation; Gretton et al. 2012a; Gretton et al. 2015; Gretton et al. 2021; Gretton et al. 2023; Gretton et al. 2023;b~~e.g., by increasing variance or for a specific kernel choice (Gretton et al., 2012a; Zhao & Meng, 2015; Cheng & Xie, 2021; Bodenham & Kawahara, 2023; Bharti et al., 2023; Gretton et al., 2012b)). While SW distance scales with $O(N \log N)$ (Nadjahi, 2021), it is difficult to make principled assessments of the computational complexity for C2ST, as it is highly dependent on the chosen classifier. ~~But~~ However, as more samples lead to larger training and test datasets, the sample size is likely to influence the compute time. Similarly, the computational complexity of computing these distances increases as the dimensionality of the data increases, with non-trivial scaling depending on the task and ~~hyperparameters chosen .~~ chosen hyperparameter. We also report the theoretical convergence of sample-based estimates for SW distance and MMD, which are subject to active research. We report bounds from recent works (Ghosal & Sen, 2019; Gretton et al., 2012a). We do not report sample complexities for C2ST and FID, as these strongly depend on the choice of classifier and embedding network, respectively.

Note that despite their differences, all presented distances are reasonably tractable in the settings of our experiments, whereas the scaling experiments might be computationally unfeasible for other distances or datasets. ~~We therefore~~ Therefore, we strongly recommend carefully considering the complexity of the measure before conducting experiments on high-dimensional or very large datasets. We report the practical computation times for our experiments in Appendix A.9.

3.5 Mode coverage properties

Mode coverage is the ability of a model to capture and generate diverse data, i.e., from multiple modes of the underlying distribution if multiple modes exist instead of from a single one (Fig. S3). If models *mode collapse* they might have learned to generate realistic but unvaried samples (Fig. S2). The community has focused on evaluation of mode coverage with different metrics driven by the development of GANs (Goodfellow et al., 2014; Gui et al., 2020; Saad et al., 2022). Empirically, in training generative models, mode coverage has been found to trade off with sample quality and speed, illustrating the generative learning trilemma (Xiao et al., 2021). All presented metrics can distinguish between a collapsed and a full distribution (Fig. 9), an empirical finding also reported in previous work (Che et al., 2016; Li et al., 2017; Deshpande et al., 2018; Borji, 2019). However, their sensitivity relies on different factors: SWD captures different modes when they are separated in one-dimensional projections, MMD depends on appropriate kernel choice, FID on expressive embeddings, and C2ST on well-trained classifiers. Other metrics used to quantify mode collapse include precision and recall (Kynkäänniemi et al., 2019).

4 Scientific applications

To demonstrate how the presented distances apply to evaluating generative models of scientific applications, we focus here on two examples: decision modeling in cognitive neuroscience and medical imaging. For each application, we used two generative models or simulators to sample synthetic data. We then compared the synthetic samples to real data (hold-out test set) using the discussed distances. To obtain baseline values for each distance, we computed distances between subsets of real data. For SW distance, MMD, and FID we anticipated values proximal to zero, while for the C2ST, we expected a value around 0.5. These baseline assessments provide a lower threshold of model fidelity to which we compared the deviation of model-generated samples.

4.1 Models of primate decision making

We explored the fidelity of two generative models in replicating primate decision times during a motion-discrimination task (Roitman & Shadlen, 2002). We evaluated two versions of a Drift-Diffusion Model (DDM; Fig. 10a) (Ratcliff, 1978), a frequently used model in cognitive neuroscience. The two versions differ with respect to the drift rate, which is the speed and direction at which evidence accumulates towards a decision, and the decision boundaries, which determine how much evidence is needed to make a decision. Specifically, the first version (DDM1) uses a drift rate that varies linearly with position and time ~~,~~ and decision boundaries that decay exponentially over time, whereas the second version (DDM2) uses a drift rate and decision boundaries that are constant over time (for details, see Section A.12). We fitted each model against empirical primate decision times with the use of the *pyDDM* toolbox (Shinn et al., 2020), generated one-dimensional synthetic datasets, and compared each dataset to the actual primate decision time distributions. While the resulting distributions of decision times are visually similar (Fig. 10b), the DDM-generated distributions DDM1 and DDM2 are noticeably broader compared to the more tightly clustered real decision times. Moreover, the DDM1 distribution appears more similar to the real distribution than that of DDM2, which is shifted towards the left. As expected, the DDM1 model more precisely mimics the

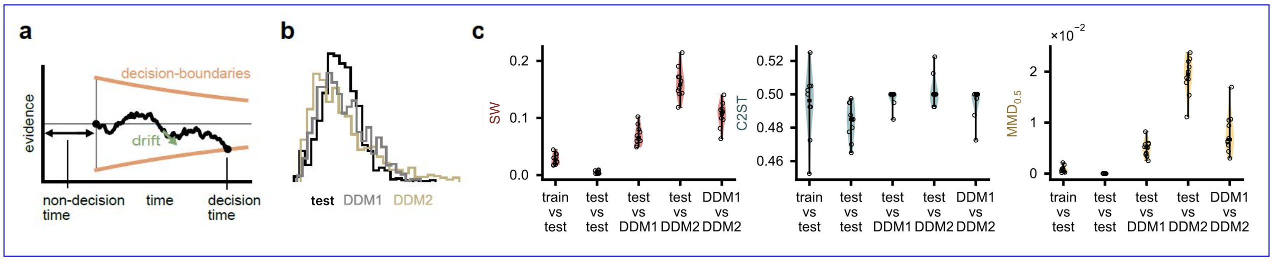

Figure 10: **Comparing models of primate decision making.** **(a)** Schematic of a ~~drift-diffusion~~ Drift-Diffusion model (DDM), a classical neuroscientific model of decision making behavior. Overall, evidence drives the model toward one of two choices (drift), but sensory and environmental noise result in random fluctuations in evidence integration (diffusion). **(b)** Distributions of primate decision times from the ~~real dataset~~ test set (black), and two fitted models of varying complexity: DDM1 (gray) and DDM2 (gold). **(c)** SW distance, C2ST, MMD (bandwidth=0.5) between subsets of the ~~three~~ generated and real data distributions. FID is not applicable in these comparisons, because the data are one-dimensional distributions. Scatter-points indicate comparisons between ten random subsets from each dataset. Thick horizontal bars indicate median values.

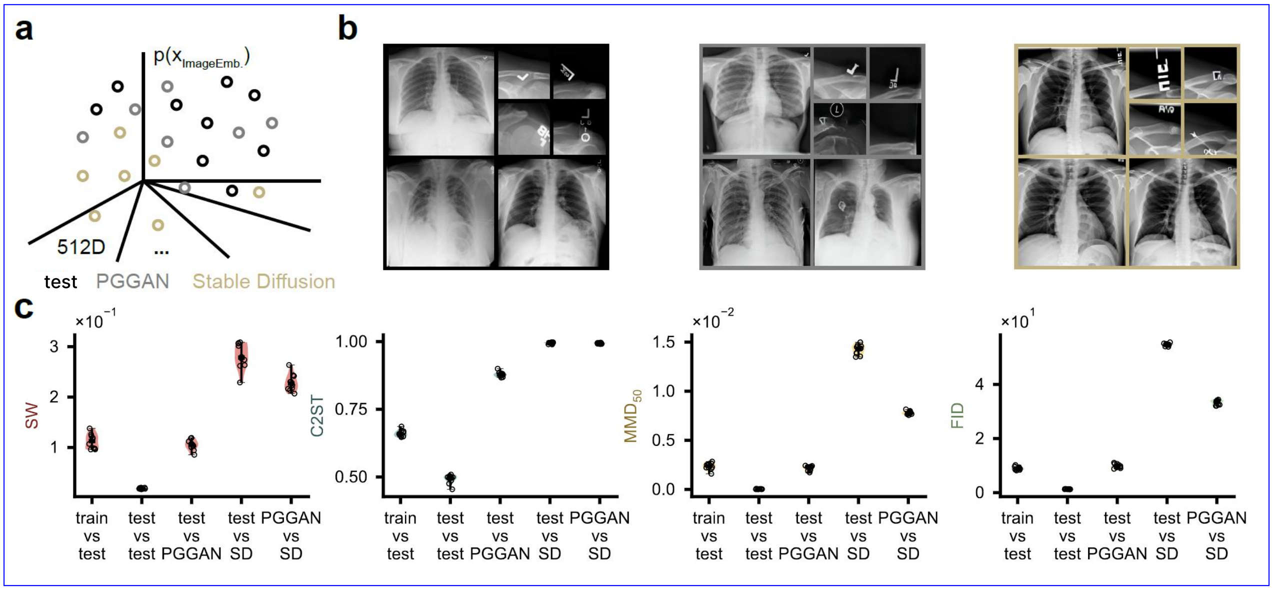

Figure 11: **Comparing generated and real X-ray images.** **(a)** Sketch of the embedded distributions of X-ray images from the three different datasets: test set of real dataset (black), Progressive Growing Generative Adversarial Network (PGGAN) (grey), and Stable Diffusion (SD) model (gold). **(b)** Examples from real and generated X-ray images. Three full-view examples from each distribution and four examples magnifying the top right corner. **(c)** SW distance, C2ST, MMD (bandwidth=50), and FID between samples of the ~~three~~ real and generated distributions of embedded X-ray images. Scatter-points indicate comparisons between ten random subsets from each dataset. Thick horizontal bars indicate median values.

real data distribution, as compared to the DDM2, across the median values of the SW distance, MMD, and C2ST distances (Fig. 10c). For C2ST, DDM1 and the real data distribution are even indistinguishable, with median C2ST values around 0.5. This suggests that SW and MMD provide a more nuanced differentiation between the models.

4.2 Chest X-ray image generation

In the second application we turned to a ~~high dimensional~~ high-dimensional example, in which we compared synthetic X-ray images generated by a Progressive Growing Generative Adversarial Network (PGGAN) model (Segal et al., 2021) and by a StableDiffusion (SD) model (Malik & Humair, 2023) to real chest X-ray images from the ChestX-ray14 dataset (Wang et al., 2017). Each image has a total dimension of 1024 × 1024 pixels.

From visual inspection, we note two observations: First, the images produced by the SD model are clearer and sharper than either the real images or those generated by the PGGAN. Second, generated images contain unrealistic artifacts that distinguish them from real X-ray images (Fig. 11b). For instance, in real images, the top often contains annotations including e.g., patient id, side of the body, or the date the X-ray was taken. These textual elements often contain artifacts or, in case of SD, are completely unrealistic. To compare these high-dimensional images, we embedded them in a 512-dimensional embedding space using the CheXzero network (Tiu et al., 2022), a CLIP (Radford et al., 2021) network fine-tuned for chest X-ray images. We opted for using this specialized network instead of the standard InceptionV3 network as it might overcome biases introduced by classification task training (Kynkäänniemi et al., 2022). As expected, samples generated by PGGAN are closer to the real data across all distances compared to SD-generated data (Fig. 11c), likely due to ~~the~~ unrealistic sharpness and more obvious textual artifacts of the SD-generated images. However, C2ST is even high between PGGAN outputs and the real data, suggesting that the high-dimensionality of the data increases the sensitivity of this measure. Taken together, our results suggest that PGGAN is more accurate in generating realistic X-ray images compared to SD.

Our findings ~~highlight~~ show that using different metrics can support different conclusions. For instance, C2ST suggests equality between DDM1 and real decision time data, whereas SW distance and MMD metrics indicate a larger difference between DDM1 and the real data. Similarly, in analyzing X-ray image generation, SW distance, MMD, and FID metrics suggest a high similarity between PGGAN-generated and real images, whereas C2ST indicates a strong difference. Thus, we want to highlight the importance of using multiple complementary distances for best results and understanding of model limitations.

5 Discussion

This work describes and explores four commonly applied sample-based distances representing different methodologies for defining statistical distance: Using low-dimensional projections (SW), obtaining a distance using classifiers (C2ST), using embeddings through kernels (MMD) or neural networks (FID). Despite their operational differences, they are all based on a fundamental concept: *simplifying complex distributions into more manageable feature representations to facilitate comparison.* Sliced distances effectively reduce multidimensional distributions to a set of one-dimensional distributions, where classical metrics are more easily applied or calculated. MMD uses kernels to (implicitly) project samples into a higher dimensional feature space, in which comparing mean values becomes more expressive. Classifier-based methods (C2ST) transform the task of distribution comparison into a classification problem; comparison is made by investigating how well a classifier can distinguish the distributions. Lastly, network-based distances, such as FID, explicitly map samples into a representative feature space and compare distributions directly within this space.

In the paragraphs below, we highlight the features and limitations of these investigated distances. Additionally, we discuss the relationships between these metrics and connect them to current related work.

Sliced Distances Sliced distances stand out for their computational efficiency in evaluating distributional discrepancies. However, when distributions differ primarily in lower-dimensional subspaces, sliced distances might not detect these subtle differences without a large number of slices (see Fig. 8c). There are approaches to reduce this effect by considering other projections than simple linear slices, as described in Section 2.1. ~~Due to its computational efficiency and differentiability, the SW distance can also be used as a loss function to train generative models (Wu et al., 2019; Deshpande et al., 2018; 2019; Liutkus et al., 2019; Vetter et al., 2024).~~ In our experiments, the metric did show convincing results and in contrast to the MMD, C2ST, and FID, SW distance does not require to choose specific hyperparameters for which results can differ drastically. ~~Although currently not extensively used in literature for evaluation, this makes the SW distance efficient, scalable, and~~

~~an objective baseline for general distribution comparisons. Yet, this also makes it less flexible to adapt to specific features of interest. Although, the~~ Yet, this also makes it less flexible to adapt to specific features of interest. The Wasserstein and SW distances are not interpretable, and admit only biased sample-based estimates. This can be a limitation for some tasks. However, Although currently not extensively used in literature for evaluation, this makes the SW distance efficient, scalable, and an objective baseline for general distribution comparisons.ue to its computational efficiency and differentiability, the SW distance is commonly used as a loss function to train generative models, such as GANs (Deshpande et al., 2018; Wu et al., 2019), Autoencoders (Wu et al., 2019), nonparametric flows (Liutkus et al., 2019), normalizing flows (Dai & Seljak, 2021), and multi-layer perceptrons (Vetter et al., 2024). Although the majority of research on sliced distances ~~focus~~ focuses on sliced *Wasserstein* metrics, slicing other metrics is also possible. For a certain subset of choices, equivalence to MMDs can be established~~ (Kolouri et al., 2019)~~(Feydy et al. (2019); Kolouri et al. (2019); Hertrich et al. (2024).

Classifier Two-Sample Test (C2ST) C2ST distinguishes itself by producing an interpretable value: classification accuracy. This characteristic makes C2ST particularly appealing for practical applications, as it is easy to explain and interpret. A notable drawback is the computational demand associated with training a classifier, which can be substantial. Moreover, C2ST's effectiveness is critically dependent on the selection and training of a suitable classifier. Interpreting results reported for C2ST requires knowledge of the classifier used and its appropriateness for the data at hand. Furthermore, automated training pipelines may encounter failures, such as when the trained classifier performs worse than chance, often due to overfitting to cross-validation folds (see also Section A.5). On the other hand, it is able to even detect subtle differences within two distributions in high dimensions. Even if there is a difference in only a single out of a thousand dimensions (for which SW distance and MMD might struggle), C2ST is able detect it (see Fig. 8c). This might be desirable, but can also be problematic. When comparing images, slight variations in a few pixels may not be visually noticeable, potentially making them unimportant to the researcher. In high-dimensional complex data, such slight variations are quite likely. Thus C2ST can be close to 1.0 in the high-dimensional setting, making it practically useless for evaluation (see Fig. 5c, 11c). The C2ST can be shown to be a MMD with a specific kernel function parameterized by the classifier (Liu et al., 2020).

MMD The Maximum Mean Discrepancy is a strong tool for comparing two groups of data by looking at their average values in a special feature space. The effectiveness of MMD largely depends on the kernel function chosen (implicitly representing the feature space), which affects how well it can spot differences between various types of data. Inappropriate kernel choice can leave the metric insensitive to subtle differences in the distribution (Gretton et al., 2012b; Sriperumbudur et al., 2009) (see Fig. 8). The MMD can be estimated efficiently and is differentiable, and thus often used as a loss function for training generative models (Arbel et al., 2019; Li et al., 2017; Bińkowski et al., 2021; Briol et al., 2019). Yet, a kernel must satisfy certain criteria, e.g., positive definiteness, making the design of new kernel functions challenging. Such constraints are relaxed for FID-like metrics, which focus on *explicit* representations of the embedding, whereas (kernel) MMD instead focuses on *implicit* representations. One advantage, however, is that the implicit embedding allows for infinite dimensional feature spaces (through characteristic kernel functions). These can be proven to be able to discriminate *any* two distinct distributions, something that is impossible through explicit representations used by the FID. Recently, Kübler et al. (2022a) proposed a method to estimate MMD via a witness function that determines MMD (Appendix A.2). This method is closely related to C2ST in that both ~~two~~ estimate a discrepancy among distributions via a classifier (Kübler et al., 2022a, Section 5).

Network-based Network-based approaches for evaluating distributions focus on the analysis of complex data, emphasizing the importance of capturing high-level, semantically meaningful features. These methods leverage neural networks to project data into a lower-dimensional, feature-rich space where traditional statistical distances can be applied more effectively. This is particularly important for tasks where the visual or semantic quality of the data is important, making them a popular choice for assessing generative models in domains such as image and text generation. The primary challenge lies in the design of suitable network architectures that can extract relevant features for accurate distribution comparison. Even more important than for the C2ST, this network must be well-established and shared which is a necessary but not sufficient criterion (Chong & Forsyth, 2020) to compare different results. While such well-established defaults exist for images (Szegedy et al., 2015a; Radford et al., 2021), this is not the case for other domains. For example, the time series generation community did not yet establish a default, and embedding ~~nets~~ networks are either trained or chosen by the authors (Smith & Smith, 2020; Jeha et al., 2021). We ~~showed~~ demonstrated that the class-sensitivity of FID (Section 3.3) ~~tends~~ often leads to model collapse~~(such as GANs), but might not necessarily reflect general~~ , as seen in GANs. However, it may not accurately reflect the overall image quality. ~~In fact~~For example, Betzalel et al. (2022); Kynkäänniemi et al. (2022); Jayasumana et al. (2023) found that relevant features sometimes can disagree with human judgment and that CLIP embeddings align more closely to what humans perceive as favorable or unfavorable. Yet, FID features have been shown to align much better with human perception than traditional metrics (Zhang et al., 2018).

Recent developments in network-based approaches include the use of Central Kernel Alignment (CKA; Cortes et al. 2012) to compute the distance between network-embedded samples. CKA scores show considerable stability when evaluated with different choices of network architectures and layers (Yang et al., 2023). Another newly introduced metric, Mauve (Pillutla et al., 2021; Liu et al., 2021; Pillutla et al., 2023), can, for instance, be used to measure how close machine-generated text is to human language using an external language model to embed the samples from each distribution. This metric uses divergence frontiers to take into account the trade-off between quality and diversity when evaluating generative models.

Closing remarks Ultimately, the choice of distance hinges on the nature of the data under consideration and the specific characteristics ~~of it~~ one aims to compare. Given a ~~specific~~ particular dataset and problem, ~~one will likely have~~ it may be necessary to look beyond the distances discussed in this paper. For ~~example~~instance, in the realm of human-centric data ~~like~~ such as images and audio, ~~the~~ perceptual indistinguishability of distributions is ~~important~~crucial (Gerhard et al., 2013; Zhang et al., 2018). Time series data, ~~characterized by its~~ with its inherent temporal structure, ~~demand metrics that account for~~ requires metrics that accommodate temporal shifts and variations ~~in a manner that does not disproportionately penalize~~ without disproportionately penalizing minor discrepancies in timing, such as Dynamic Time Warping (Müller, 2007) ~~, or by using frequency information~~or frequency-based methods (Hess et al., 2023). ~~In general however, irrespective~~ Additional recent works propose new distances based on fields such as topology (Barannikov et al. 2021).

In general, regardless of the specific ~~use-case~~use case, it is advisable to use multiple ~~distances in order~~ distance measures to obtain a ~~full picture, as using a single distances individually could support~~ comprehensive view, as relying on a single measure may lead to competing conclusions about the model ~~that is to be evaluated~~under evaluation.

Throughout this paper, we have explained and analyzed four approaches ~~of~~ to measuring statistical distance. While this ~~is~~ represents only a small subset of all possible ~~distances available~~measures, we hope to have provided the foundational knowledge ~~with which researchers can~~ researchers need to find, understand, and interpret statistical distances ~~specific~~ relevant to their own scientific ~~application~~applications.

Code availability

All code for replicating and running our analysis is available at: https://anonymous.4open.science/r/tmlr-anonymized-6AC5/.

References

Michael Arbel, Anna Korba, Adil Salim, and Arthur Gretton. Maximum mean discrepancy gradient flow. *arXiv preprint arXiv:1906.04370*, 2019.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Serguei Barannikov, Ilya Trofimov, Grigorii Sotnikov, Ekaterina Trimbach, Alexander Korotin, Alexander Filippov, and Evgeny Burnaev. Manifold topology divergence: a framework for comparing data manifolds. *Advances in neural information processing systems*, 34:7294–7305, 2021.

Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

Michéle Basseville. Divergence measures for statistical data processing - an annotated bibliography. *Signal Processing*, 2013.

Espen Bernton, Pierre E Jacob, Mathieu Gerber, and Christian P Robert. On parameter estimation with the Wasserstein distance. *Information and Inference: A journal of the IMA*, 2019.

Eyal Betzalel, Coby Penso, Aviv Navon, and Ethan Fetaya. A study on the evaluation of generative models. *arXiv preprint arXiv:2206.10935*, 2022.

Ayush Bharti, Masha Naslidnyk, Oscar Key, Samuel Kaski, and Francois-Xavier Briol. Optimally-weighted estimators of the maximum mean discrepancy for likelihood-free inference. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Felix Biggs, Antonin Schrab, and Arthur Gretton. MMD-Fuse: Learning and combining kernels for two-sample testing without data splitting. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

Christopher M Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. *arXiv preprint arXiv:1801.01401*, 2021.

Dean A. Bodenham and Yoshinobu Kawahara. euMMD: Efficiently computing the MMD two-sample test statistic for univariate data. *Statistics and Computing*, 2023.

Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G Willcocks. Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, Jan 2015. ISSN 1573-7683. doi: 10.1007/s10851-014-0506-3. URL `https://doi.org/10.1007/s10851-014-0506-3`.

Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 2006.

Ali Borji. Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding*, 2019.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control.* John Wiley & Sons, 2015.

Manuel Brenner, Florian Hess, Jonas M. Mikhaeil, Leonard Bereska, Zahra Monfared, Po-Chen Kuo, and Daniel Durstewitz. Tractable dendritic RNNs for reconstructing nonlinear dynamical systems. *arXiv preprint arXiv:2207.02542*, 2022.

Francois-Xavier Briol, Alessandro Barp, Andrew B. Duncan, and Mark Girolami. Statistical inference for generative models with maximum mean discrepancy. *arXiv preprint arXiv:1906.05944*, 2019.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020.

Elsa Cazelles, Arnaud Robert, and Felipe Tobar. The Wasserstein-Fourier distance for stationary time series. *arXiv preprint arXiv:1912.05509*, 2020.

Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv [cs.LG]*, December 2016.

Xiuyuan Cheng and Alexander Cloninger. Classification logit two-sample testing by neural networks for differentiating near manifold densities. *IEEE Transactions on Information Theory*, 2022.

Xiuyuan Cheng and Yao Xie. Neural tangent kernel maximum mean discrepancy. In *Advances in Neural Information Processing Systems*, 2021.

Min Jin Chong and David Forsyth. Effectively unbiased FID and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.

Biwei Dai and Uros Seljak. Sliced iterative normalizing flows. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2352–2364. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/dai21a.html`.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 2009.

Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the Sliced-Wasserstein distance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. Max-sliced Wasserstein distance and its use for GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. GENIE: Higher-order denoising diffusion solvers. In *Advances in Neural Information Processing Systems*, 2022.

Chao Du, Tianbo Li, Tianyu Pang, Shuicheng Yan, and Min Lin. Nonparametric generative modeling with conditional Sliced-Wasserstein flows. *arXiv preprint arXiv:2305.02164*, 2023.

Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. Reconstructing computational system dynamics from neural data with recurrent neural networks. *Nature Reviews Neuroscience*, 2023.

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

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2681–2690. PMLR, 2019.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3):707–738, Aug 2015. ISSN 1432-2064. doi: 10.1007/s00440-014-0583-7. URL `https://doi.org/10.1007/s00440-014-0583-7`.

Jerome H. Friedman. On multivariate goodness of fit and two sample testing. *eConf*, 2003.

Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Bharath K. Sriperumbudur. Characteristic kernels on groups and semigroups. In *Advances in Neural Information Processing Systems*, 2008.

Ruize Gao, Feng Liu, Jingfeng Zhang, Bo Han, Tongliang Liu, Gang Niu, and Masashi Sugiyama. Maximum mean discrepancy test is aware of adversarial attacks. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Thomas Gärtner. A survey of kernels for structured data. *ACM SIGKDD explorations newsletter*, 2003.

Holly E. Gerhard, Felix A. Wichmann, and Matthias Bethge. How sensitive is the human visual system to the local statistics of natural images? *PLOS Computational Biology*, 2013.

Promit Ghosal and Bodhisattva Sen. Multivariate ranks and quantiles using optimal transport: Consistency, rates and nonparametric testing. *The Annals of Statistics*, 2019. URL `https://api.semanticscholar.org/CorpusID:233740353`.

Alison L. Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International Statistical Review / Revue Internationale de Statistique*, 2002.

Ziv Goldfeld and Kristjan Greenewald. Sliced mutual information: A scalable measure of statistical dependence. *Advances in Neural Information Processing Systems*, 2021.

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

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

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

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10696–10706, 2022.

Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv [cs.LG]*, January 2020.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

S. Helgason. *The Radon Transform*. Progress in Mathematics - Birkhäuser. Birkhäuser, 1980. ISBN 9783764330064. URL `https://books.google.de/books?id=9pCpAAAAIAAJ`.

Johannes Hertrich, Christian Wald, Fabian Altekrüger, and Paul Hagemann. Generative sliced MMD flows with riesz kernels. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=VdkGRV1vcf`.

Florian Hess, Zahra Monfared, Manuel Brenner, and Daniel Durstewitz. Generalized teacher forcing for learning chaotic dynamics. *arXiv preprint arXiv:2306.04406*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, 2017.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, and Bernhard Nessler. FID score for PyTorch, 2018a.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, and Bernhard Nessler. Two time-scale update rule for training GANs, 2018b.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 2006.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.

A. L. Hodgkin and A. F. Huxley. Currents carried by sodium and potassium ions through the membrane of the giant axon of Loligo. *The journal of Physiology*, 1952.

Mozes Jacobs, Bingni W. Brunton, Steven L. Brunton, J. Nathan Kutz, and Ryan V. Raut. HyperSINDy: Deep generative modeling of nonlinear stochastic governing equations. *arXiv preprint arXiv:2310.04832*, 2023.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking FID: Towards a better evaluation metric for image generation. *arXiv preprint arXiv:2401.09603*, 2023.

Paul Jeha, Michael Bohlke-Schneider, Pedro Mercado, Shubham Kapoor, Rajbir Singh Nirwan, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. PSA-GAN: Progressive self attention GANs for synthetic time series. In *International Conference on Learning Representations*, 2021.

Wittawat Jitkrittum, Zoltán Szabó, Kacper P Chwialkowski, and Arthur Gretton. Interpretable distribution features with maximum testing power. In *Advances in Neural Information Processing Systems*, 2016.

Ilmun Kim, Ann B. Lee, and Jing Lei. Global and local two-sample tests via regression. *arXiv preprint arXiv:1812.08927*, 2019.

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

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized Sliced-Wasserstein distances. *Advances in Neural Information Processing Systems*, 2019.

Jonas M. Kübler, Wittawat Jitkrittum, Bernhard Schölkopf, and Krikamol Muandet. A witness two-sample test. In *International Conference on Artificial Intelligence and Statistics*, 2022a.

Jonas M. Kübler, Vincent Stimper, Simon Buchholz, Krikamol Muandet, and Bernhard Schölkopf. Automl two-sample test. In *Advances in Neural Information Processing Systems*, 2022b.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 1955.

Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in Fréchet inception distance. *arXiv preprint arXiv:2203.06026*, 2022.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *arXiv [stat.ML]*, April 2019.

Christina Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: A string kernel for svm protein classification. In *Biocomputing*, 2001.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD-GAN: Towards deeper understanding of moment matching network. *Advances in Neural Information Processing Systems*, 2017.

Yujia Li, Kevin Swersky, and Richard Zemel. Generative moment matching networks. *arXiv preprint arXiv:1502.02761*, 2015.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J. Sutherland. Learning deep kernels for non-parametric two-sample tests. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Lang Liu, Krishna Pillutla, Sean Welleck, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. Divergence frontiers for generative models: Sample complexity, quantization effects, and frontier integrals. *Advances in Neural Information Processing Systems*, 34:12930–12942, 2021.

Antoine Liutkus, Umut Şimşekli, Szymon Majewski, Alain Durmus, and Fabian-Robert Stöter. Sliced-Wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. *arXiv preprint arXiv:1806.08141*, 2019.

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

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2002.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*, 2016.

Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking simulation-based inference. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. *arXiv preprint arXiv:2201.09865*, 2022.

Muhammad Danyal Malik and Danish Humair. Evaluating the feasibility of using generative models to generate chest X-ray data. *arXiv preprint arXiv:2305.18927*, 2023.

Mohamed Marouf, Pierre Machart, Vikas Bansal, Christoph Kilian, Daniel S Magruder, Christian F Krebs, and Stefan Bonn. Realistic in silico generation and augmentation of single-cell rna-seq data using generative adversarial networks. *Nature communications*, 2020.

Alexander Mathiasen and Frederik Hvilshøj. Backpropagating through fréchet inception distance. *arXiv preprint arXiv:2009.14075*, 2021.

Midjourney. Midjourney, 2022. URL `https://www.midjourney.com/home/`. Accessed: 2024-06-06.

George A Miller. WordNet: a lexical database for english. *Communications of the ACM*, 1995.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 2017.

Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, 2007.

Kimia Nadjahi. *Sliced-Wasserstein distance for large-scale machine learning: theory, methodology and extensions*. PhD thesis, Institut polytechnique de Paris, 2021.

Kimia Nadjahi, Alain Durmus, Lénaïc Chizat, Soheil Kolouri, Shahin Shahrampour, and Umut Simsekli. Statistical and topological properties of sliced probability divergences. *Advances in Neural Information Processing Systems*, 2020.

Khai Nguyen and Nhat Ho. Sliced wasserstein estimation with control variates, 2024.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Victor M Panaretos and Yoav Zemel. Statistical aspects of Wasserstein distances. *Annual review of statistics and its application*, 2019.

Teodora Pandeva, Tim Bakker, Christian A Naesseth, and Patrick Forré. E-valuating classifier two-sample tests. *arXiv preprint arXiv:2210.13027*, 2022.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 2021.

Tamás Papp and Chris Sherlock. Bounds on wasserstein distances between continuous distributions using independent samples, 2022.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2011.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Center for Research in Economics and Statistics Working Papers*, 2017.

Jonathan W Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, EJ Chichilnisky, and Eero P Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 2008.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

Krishna Pillutla, Lang Liu, John Thickstun, Sean Welleck, Swabha Swayamdipta, Rowan Zellers, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. Mauve scores for generative models: Theory and practice. *Journal of Machine Learning Research*, 24(356):1–92, 2023.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, 2021.

Sebastian Raschka. An overview of general performance metrics of binary classifier systems. *arXiv preprint arXiv:1410.5330*, 2014.

Roger Ratcliff. A theory of memory retrieval. *Psychological review*, 1978.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Jamie D Roitman and Michael N Shadlen. Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task. *Journal of Neuroscience*, 2002.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022a.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022b.

Cédric Rommel, Joseph Paillard, Thomas Moreau, and Alexandre Gramfort. Data augmentation for learning predictive models on EEG: a systematic comparison. *journal of Neural Engineering*, 2022.

Muhammad Muneeb Saad, Ruairi O'Reilly, and Mubashir Husain Rehmani. A survey on training challenges in generative adversarial networks for biomedical image analysis. *arXiv [cs.LG]*, January 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *arXiv preprint arXiv:1606.03498*, 2016.

Antonin Schrab, Ilmun Kim, Mélisande Albert, Béatrice Laurent, Benjamin Guedj, and Arthur Gretton. MMD aggregated two-sample test. *Journal of Machine Learning Research*, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Bradley Segal, David M. Rubin, Grace Rubin, and Adam Pantanowitz. Evaluating the clinical realism of synthetic chest X-rays generated using progressively growing GANs. *SN Computer Science*, 2021.

Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5), October 2013. ISSN 0090-5364. doi: 10.1214/13-aos1140. URL `http://dx.doi.org/10.1214/13-AOS1140`.

Maxwell Shinn, Norman H Lam, and John D Murray. A flexible framework for simulating and fitting generalized drift-diffusion models. *ELife*, 2020.

Kaleb E Smith and Anthony O Smith. Conditional GAN for timeseries generation. *arXiv preprint arXiv:2006.16477*, 2020.

Rachel S Somerville and Romeel Davé. Physical models of galaxy formation in a cosmological framework. *Annual Review of Astronomy and Astrophysics*, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023.

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

Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R.G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 2011.

Akash Srivastava, Kai Xu, Michael U. Gutmann, and Charles Sutton. Generative ratio matching networks, 2020.

Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Charu Bai Reddy, Matteo Carandini, and Kenneth D. Harris. Spontaneous behaviors drive multidimensional, brainwide activity. *Science*, 2019.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, USA, 1st edition, 2012. ISBN 0521190177.

Danica J. Sutherland. Maximum mean discrepancy (distance distribution). Cross Validated, 2019. URL `https://stats.stackexchange.com/q/276618`.

Danica J. Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alex Smola, and Arthur Gretton. Generative models and model criticism via optimized maximum mean discrepancy. In *International Conference on Learning Representations*, 2021.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015a.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015b.

Gábor J. Székely and Maria L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013. ISSN 0378-3758. doi: https://doi.org/10.1016/j.jspi.2013.03.018. URL `https://www.sciencedirect.com/science/article/pii/S0378375813000633`.

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

Michalis K. Titsias and Francisco Ruiz. Unbiased implicit variational inference. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 167–176. PMLR, 16–18 Apr 2019. URL `https://proceedings.mlr.press/v89/titsias19a.html`.

Ekin Tiu, Ellie Talius, Pujan Patel, Curtis P Langlotz, Andrew Y Ng, and Pranav Rajpurkar. Expert-level detection of pathologies from unannotated chest X-ray images via self-supervised learning. *Nature Biomedical Engineering*, 2022.

Fabio Urbina, Christopher T Lowden, J Christopher Culberson, and Sean Ekins. MegaSyn: integrating generative molecular design, automated analog designer, and synthetic viability prediction. *ACS omega*, 2022.

Cyril Vallez, Andrei Kucharavy, and Ljiljana Dolamic. Needle in a haystack, fast: Benchmarking image perceptual similarity metrics at scale. *arXiv preprint arXiv:2206.00282*, 2022.

Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, 2016.

Julius Vetter, Jakob H. Macke, and Richard Gao. Generating realistic neurophysiological time series with denoising diffusion probabilistic models. *bioRxiv*, 2023.

Julius Vetter, Guy Moss, Cornelius Schröder, Richard Gao, and Jakob H. Macke. Sourcerer: Sample-based maximum entropy source distribution estimation. *arXiv preprint arXiv:2402.07808*, 2024.

S. V. N. Vishwanathan, Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 2010.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. ChestX-Ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Wukong. Wukong, 2022. URL `https://xihe.mindspore.cn/modelzoo/wukong`. Accessed: 2024-06-06.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. *arXiv [cs.LG]*, December 2021.

Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755*, 2018.

Ceyuan Yang, Yichi Zhang, Qingyan Bai, Yujun Shen, Bo Dai, et al. Revisiting the evaluation of image synthesis with gans. *Advances in Neural Information Processing Systems*, 36:9518–9542, 2023.

Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *arXiv preprint arXiv:2305.10435*, 2023.

Mingxuan Yi and Song Liu. Sliced-Wasserstein variational inference. In *Proceedings of The 14th Asian Conference on Machine Learning*, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

Ji Zhao and Deyu Meng. FastMMD: Ensemble of circular discrepancy for efficient two-sample test. *Neural Computation*, 2015.

Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. Genimage: A million-scale benchmark for detecting ai-generated image, 2023.

A Appendix

A.1 Generative Models in Science

Table S1: **Example generative models in science.** Simulators include mechanistic models, DNN based models e.g. VAEs, GANs, Diffusion models. This is not an exhaustive list regarding disciplines using generative models nor generative models used in the listed disciplines.

	ML model	Simulator
Biology		
- single cell sequencing	[1–5]	[6–11]
- cellular biology [12]	[13–15]	[16–19]
Geoscience		
- ice flow modelling	[20–24]	[25–27]
- Numerical weather prediction	[28–30]	[31–33]
Chemistry		
- molecule generation [34]	[35–45]	
Astronomy		
- astronomical images	[46–48]	[49; 50]

A.2 Details about Maximum Mean Discrepancy

Here we provide different formulations and examples of MMD.

Definition A.1 (Feature Map Definition of MMD)

$$\mathsf{MMD}^2[\phi, p_1, p_2] = \|\mathbb{E}_{p_1(x)}[\phi(x)] - \mathbb{E}_{p_2(y)}[\phi(y)]\|^2_{\mathcal{H}}, \tag{5}$$

where $p_1(x)$ and $p_2(y)$ are the probability distributions of random variables $x, y \in \mathcal{X}$, and $\phi : \mathcal{X} \to \mathcal{H}$.

Generally $\mathcal{X}$ and $\mathcal{H}$ are defined as a topological space, and the reproducing kernel Hilbert space (RKHS), respectively, but readers can simply think of the euclidean space $\mathbb{R}^N$ for the first examples in the main text.

For the *identity feature map* $\phi^{(1)} : \mathbb{R} \to \mathbb{R}, \phi^{(1)}(x) = x$, MMD can be computed as

$$\begin{aligned}
\mathsf{MMD}^2[\phi^{(1)}, p_1, p_2] &= \|\mathbb{E}_{p_1(x)}[x] - \mathbb{E}_{p_2(y)}[y]\|^2_{\mathbb{R}} \\
&= (\mathbb{E}_{p_1(x)}[x] - \mathbb{E}_{p_2(y)}[y])^2 \\
\mathsf{MMD}[\phi^{(1)}, p_1, p_2] &= |\mu_{p_1} - \mu_{p_2}|.
\end{aligned}$$

And for the *quadratic polynomial feature map* $\phi^{(2)} : \mathbb{R} \to \mathbb{R}^2, \phi^{(2)}(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$, MMD can be computed as

$$\begin{aligned}
\mathsf{MMD}^2[\phi^{(2)}, p_1, p_2] &= \|\mathbb{E}_{p_1(x)}[\begin{bmatrix} x \\ x^2 \end{bmatrix}] - \mathbb{E}_{p_2(y)}[\begin{bmatrix} y \\ y^2 \end{bmatrix}]\|^2_{\mathbb{R}} \\
&= \|\begin{bmatrix} \mu_{p_1} \\ \mu^2_{p_1} + \sigma^2_{p_1} \end{bmatrix} - \begin{bmatrix} \mu_{p_2} \\ \mu^2_{p_2} + \sigma^2_{p_2} \end{bmatrix}\|^2_{\mathbb{R}} \\
\mathsf{MMD}^2[\phi^{(2)}, p_1, p_2] &= (\mu_{p_1} - \mu_{p_2})^2 + (\mu^2_{p_1} + \sigma^2_{p_1} - \mu^2_{p_2} - \sigma^2_{p_2})^2.
\end{aligned}$$

Definition A.2 (Kernel Definition of MMD)

$$\begin{aligned}
\mathsf{MMD}^2[\phi, p_1, p_2] &= \|\mathbb{E}_{p_1(x)}[\phi(x)] - \mathbb{E}_{p_2(y)}[\phi(y)]\|^2_{\mathcal{H}} \\
&= \langle \mathbb{E}_{p_1(x)}[\phi(x)], \mathbb{E}_{p_1(x)}[\phi(x)] \rangle_{\mathcal{H}} + \langle \mathbb{E}_{p_2(y)}[\phi(y)], \mathbb{E}_{p_2(y)}[\phi(y)] \rangle_{\mathcal{H}} - 2\langle \mathbb{E}_{p_1(x)}[\phi(x)], \mathbb{E}_{p_2(y)}[\phi(y)] \rangle_{\mathcal{H}} \\
&= \mathbb{E}_{p_1(x), p'_1(x')}[\langle \phi(x), \phi(x') \rangle_{\mathcal{H}}] + \mathbb{E}_{p_2(y), p'_2(y')}[\langle \phi(y), \phi(y') \rangle_{\mathcal{H}}] - 2\mathbb{E}_{p_1(x), p_2(y)}[\langle \phi(x), \phi(y) \rangle_{\mathcal{H}}] \\
\mathsf{MMD}^2[k, p_1, p_2] &= \mathbb{E}_{p_1(x), p'_1(x')}[k(x, x')] + \mathbb{E}_{p_2(y), p'_2(y')}[k(y, y')] - 2\mathbb{E}_{p_1(x), p_2(y)}[k(x, y)].
\end{aligned}$$

The definition of MMD can be rewritten through the notion *kernel mean embedding*. For given distribution $p(x)$, the kernel mean embedding $\mathbb{E}_{p(x)}[k(x,u)] \in \mathcal{H}$ is an element in RKHS that satisfies $\langle \mathbb{E}_{p(x)}[k(x,u)], f(u) \rangle_{\mathcal{H}} = \mathbb{E}_{p(x)}[f(x)]$ for any $f \in \mathcal{H}$ with argument $u \in \mathcal{X}$. The embedding $\mathbb{E}_{p(x)}[k(x,u)]$ is known to be determined uniquely if a corresponding kernel is bounded, *i.e.*, $\|k(x,x')\|_{\mathcal{H}} < \infty$ for any x. Then, as shown in Gretton et al. (2012a), MMD^2 can be represented as

$$\mathsf{MMD}^2[k, p_1, p_2] = \|\mathbb{E}_{p_1(x)}[k(x,u)] - \mathbb{E}_{p_2(y)}[k(y,u)]\|^2_{\mathcal{H}}.$$

MMD can also be defined more generally as the integral probability metric.

Definition A.3 (Supremum Definition of MMD)

$$\mathsf{MMD}[\mathcal{F}, p_1, p_2] = \sup_{f \in \mathcal{F}} (\mathbb{E}_{p_1(x)}[f(x)] - \mathbb{E}_{p_2(y)}[f(y)]). \tag{6}$$

Here, $\mathcal{F}$ is a class of functions $f : \mathcal{X} \to \mathbb{R}$. Where we take $\mathcal{F}$ as the unit ball in an RKHS $\mathcal{H}$ with associated kernel $k(x,x')$ (Gretton et al., 2012a), the function that attains supremum (the witness function) is

$$f(u) = \frac{\mathbb{E}_{p_1(x)}[k(x,u)] - \mathbb{E}_{p_2(y)}[k(y,u)]}{\|\mathbb{E}_{p_1(x)}[k(x,u)] - \mathbb{E}_{p_2(y)}[k(y,u)]\|_{\mathcal{H}}}.$$

Assigning $f(u)$ into (Eq. (6)), we have

$$\begin{aligned}
\mathsf{MMD}^2[\mathcal{F}, p_1, p_2] &= \Big(\sup_{f \in \mathcal{F}} (\mathbb{E}_{p_1(x')}[f(x')] - \mathbb{E}_{p_2(y')}[f(y')]) \Big)^2 \\
&= \Big(\frac{\mathbb{E}_{p_1(x),p_1'(x')}[k(x,x')] + \mathbb{E}_{p_2(y),p_2'(y')}[k(y,y')] - 2\mathbb{E}_{p_1(x),p_2(y)}[k(x,y)]}{\|\mathbb{E}_{p(x)}[k(x,u)] - \mathbb{E}_{p_2(y)}[k(y,u)]\|_{\mathcal{H}}} \Big)^2 \\
&= \Big(\frac{\|\mathbb{E}_{p_1(x)}[k(x,u)] - \mathbb{E}_{p_2(y)}[k(y,u)]\|^2_{\mathcal{H}}}{\|\mathbb{E}_{p_1(x)}[k(x,u)] - \mathbb{E}_{p_2(y)}[k(y,u)]\|_{\mathcal{H}}} \Big)^2 \\
&= \|\mathbb{E}_{p_1(x)}[k(x,u)] - \mathbb{E}_{p_2(y)}[k(y,u)]\|^2_{\mathcal{H}},
\end{aligned}$$

which is equal to the kernel definition of MMD.

Definition A.4 (Characteristic Kernel) *A kernel is called* characteristic *when the kernel mean embedding*

$$p(x) \mapsto f(u) = \mathbb{E}_{p(x)}[k(x,u)] \in \mathcal{H}$$

is injective (Sriperumbudur et al., 2011; Fukumizu et al., 2008).

This means that, if a characteristic kernel is used, the embedding into the RKHS can uniquely preserve all information about a distribution. In our evaluation, we utilize the Gaussian kernel, one of the well-known characteristic kernels. Another example of a characteristic kernel is the Laplacian kernel, which is defined by

$$k(x,x') := \exp\big(-\beta|x - x'|\big).$$

Note that linear and polynomial kernels are not characteristic, while they are quite popular in natural language processing.

A.3 ~~Details about~~ Formal Definition of Wasserstein and Sliced-Wasserstein ~~distances~~Distance

~~Formally, the Wasserstein distance is described in measure-theoretic terms. We first state this definition, and then provide an accessible interpretation in the common case that the measures have well-defined probability density functions.~~

Let (M, ρ) be a Polish space, $\mu, \nu \in P(M)$ be two probability measures, and let $q \in [1, +\infty)$. Then the Wasserstein-q distance between μ and ν is defined as

$$W_q(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \mathbb{E}_{x_1, x_2 \sim \gamma} \rho(x, y)^q \right)^{1/q},$$

where $\Gamma(\mu, \nu)$ is the set of all couplings between μ, ν.

~~Consider the case where $M \subseteq \mathbb{R}^d$, ρ is the q-norm $||\cdot||_q$, and μ and ν have well-defined probability density functions p_1 and p_2 respectively~~The Sliced-Wasserstein distance is closely connected to the Radon transform (Helgason, 1980). We direct readers to Bonneel et al. (2015) for details. Here we provide a shorter definition following Definition 2.9 of Nadjahi (2021).

Suppose that $M \subset \mathbb{R}^d$, and denote by $\mathbb{S}^{d-1} = \{\theta \in \mathbb{R}^d : ||\theta||_2 = 1\}$ be the unit sphere with respect to the Euclidean norm. Let $u^* : X \to \mathbb{R}$ be a the linear form given by $u^*(x) = \langle u, x \rangle$, and $q \in [1, +\infty)$. Then the ~~Wasserstein-~~Sliced-Wasserstein distance (of order q~~norm can be written as~~) is defined for any measures $\mu, \nu \in P_q(X)$ as

$$\underline{W}SW_q(\underline{p_1}\mu, \underline{p_2}\nu) = \underline{\inf_{\gamma \sim \Gamma(p_1, p_2)}} \left(\underline{\mathbb{E}_{x_1, x_2 \sim \gamma} ||x_1 - x_2||} \int_{\mathbb{S}^{d-1}} W_q^q(u^*_{\#}\mu, u^*_{\#}\nu) d\mathcal{U}(u) \right) \underline{\tfrac{1}{q}}{}^{1/q},$$

~~where $\Gamma(p_1, p_2)$ is the set of all couplings, that is all possible "transportation plans", between p_1, p_2. $\gamma \in \Gamma(p_1, p_2)$ is a joint distribution over (x_1, x_2) with respective marginals p_1 and p_2 over x_1 and x_2.~~ Where $\mathcal{U}$ is the uniform distribution on $\mathbb{S}^{d-1}$, and for any $u \in \mathbb{S}^{d-1}$, $u^*_{\#}$ denotes the push-forward operator of u^*.

~~The Sliced-Wasserstein distanceis similarly defined in measure-theoretic terms for the measures~~ As for the Wasserstein distance, the definition becomes more intuitive in the case where μ and ν ~~. We refer the reader to Nadjahi (2021) for details. In the less general case described above, we can similarly provide a more intuitive definition. In~~ admit the probability density functions (p_1 and p_2 respectively). In particular, the random projection ~~directions described in 2.1~~ u are uniformly random vectors ~~$u \in \mathbb{S}^{d-1}$, the unit sphere in $\mathbb{R}^d$. Projecting~~ in $\mathbb{S}^{d-1}$. Therefore, projecting the distributions p_1 and p_2 onto u induces one-dimensional distributions p_1^u and p_2^u with samples $u^\intercal x_i$, where $x_1 \sim p_1$ and $x_2 \sim p_2$. The Sliced-Wasserstein-q distance can then be written as

$$\underline{SW_q(p_1, p_2) = \mathbb{E}_{u \sim \mathcal{U}(\mathbb{S}^{d-1})}[W_q(p_1^u, p_2^u)],}$$

$$SW_q(p_1, p_2) = \left(\mathbb{E}_{u \sim \mathcal{U}(\mathbb{S}^{d-1})}[W_q^q(p_1^u, p_2^u)] \right)^{1/q}. \tag{7}$$

~~where $\mathcal{U}(\mathbb{S}^{d-1})$ is the uniform distribution over vectors on the unit sphere $\mathbb{S}^{d-1}$.~~

A.4 Dependence of SW Distance on Number of Projections

All SW distance experiments in the main text were performed with the SW distance approximate with 100 random projections to approximate the expectation Eq. (7). Here, we additionally show the dependence of the SW distance approximation with finite projections on the d-dim Gaussian example (see Sec.±3), for $d \in \{10, 100, 1000\}$ (Fig S1). While the sample-based approximation to the analytic 1-dimensional Wasserstein distance is biased (Sec. 2.1), the Monte Carlo approximation to the expectation Eq. (7) is an unbiased estimate of the sample-based Wasserstein distance. We observe that for very high (1000)-dimensional distributions, the SW distance estimate converges quickly, and the choice of 100 random projections for the computation of the SW distance is appropriate.

A.5 C2ST scores below 0.5

In practice the C2ST score can sometimes turn out to be below .5. That is, the trained classifier performs systematically worse than a random classifier. A potential reason for this effect is the existence of near duplicates or copies between the two different sets. Before training the classifier, these duplicates are assigned opposite class labels. When a given pair of such duplicates is then split into one that belongs to the training

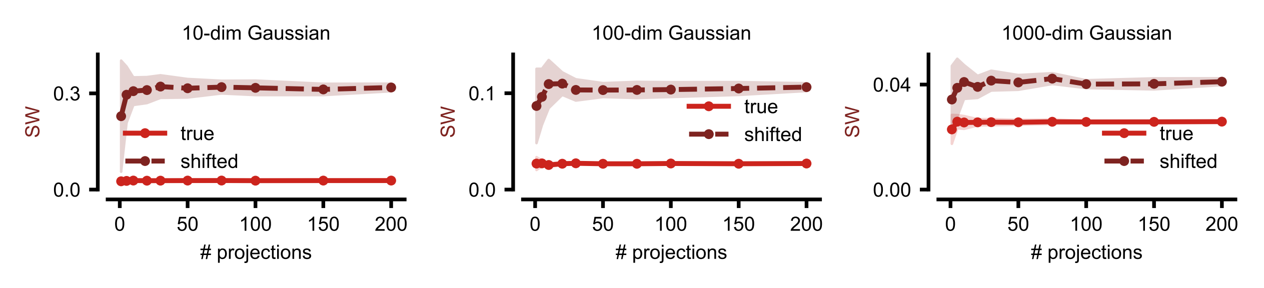

Figure S1: **The SW distance estimate is not strongly sensitive to the number of random projections.** We compare the SW distance estimate for the $\{10, 100, 1000\}$-dimensional Gaussian task with 1 shifted dimension (Sec. 3) as we increase the number of random projections used in the estimation. As the number of projections increases, the variance of the SW distance estimate decreases, but across all dimensionalities considered, the SW distance estimate has converged by 100 random projections.

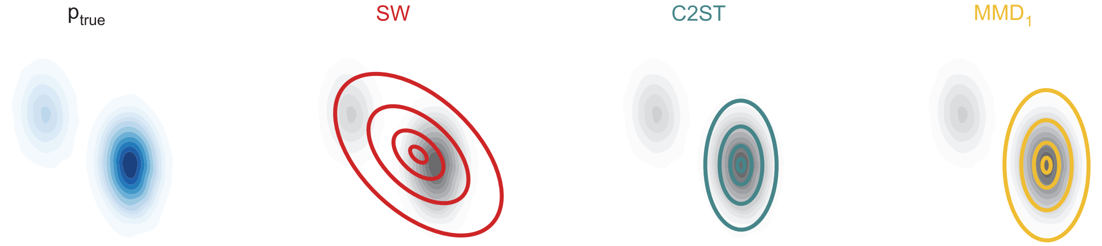

Figure S2: **Trade-offs illustrated through optimization of a miss-specified model.** We fitted a misspecified model (a two-dimensional Gaussian) by using different distances to a multi-modal distribution (similar to Theis et al. 2016). Note, that for a well-specified model each distance would give a perfect fit (Section A.6.2). In the optimization we minimized the SW distance, the C2ST classifiability, and the MMD with a Gaussian Kernel (details in Section A.6.1). Plotted are the contour lines of .25, .75, 1, and 2.5 standard deviations of the fitted Gaussians. The model optimised with SW is *mass-covering*: it covers both modes and therefore also assigns density to low-density regions of the true distribution, thus producing varied, but potentially unlikely samples. The models optimised with C2ST and MMD are *mode-seeking*: they have high densities only in the largest mode of the true distribution, and thus produces likely, but unvaried samples.

set and one that belongs to the test set, the classifier is biased towards predicting the wrong class for the duplicate in the test set. This effect is particularly noticeable if the classifier was not carefully regularized during training, and thus memorized the class label of the duplicate in the training set.

A.6 Sliced-Wasserstein, MMD and C2ST as optimization target

A.6.1 Fitting a Gaussian with gradient descent

We provide an illustrative example of which distributions are obtained when using Wasserstein, MMD and C2ST distances as a goodness of fit criterion, for both a miss-specified example, in Fig. S2 and well-specified example Fig. S3. We can see that in the miss-specified example different distances make different trade-offs, for example whether they are mode-seeking, and produce likely but unvaried samples, or are mode-covering, where they produce varied, but also potentially unlikely samples.

For Wasserstein, optimisation has been studied more formally in previous work (Bernton et al., 2019; Yi & Liu, 2023). Wasserstein was used as training objective in Arjovsky et al. (2017) and MMD was used as training objective in Bińkowski et al. (2021); Dziugaite et al. (2015); Li et al. (2015). Optimizing the C2ST classifier at the same time as the parameters of our generative model is similar to training a GAN (Goodfellow et al., 2014), but for simplicity we instead optimized for the closed-form optimal C2ST as for this toy example we have access to true densities. While FID can be used as optimization target in principle (Mathiasen & Hvilshøj, 2021), its applicability to our toy example here is less obvious, so we excluded it here.

In order to fit the (miss-specified) Gaussian model,

$$p(x) = \mathcal{N}(\mu, \Sigma)$$

to the ground truth distribution p_{true}, which is a mixture of Gaussians, we proceed as follows. Let CC^T denote the Cholesky decomposition of Σ. We compute gradients with respect to μ and C by using the reparameterization trick; by generating samples as $\mu + C\epsilon$, with $\epsilon \sim \mathcal{N}(0, \mathsf{I})$.

For Wasserstein we used as loss the Sliced Wasserstein distance, for MMD, we used a Gaussian Kernel with bandwidth set according the median heuristic.

For C2ST, we can evaluate the probability densities of samples from both the learned Gaussian and the ground truth mixture of Gaussians, so we minimize the accuracy of the closed-form optimal classifier. For each sample, we evaluate the log-probability density of the sample under each distribution, softmax the two resulting values, and use those as the classifier predicted probabilities. We then use binary cross-entropy as the loss function.

We used the ADAM optimizer (Kingma & Ba, 2015), with learning rate=0.01 and default momentums, using 2500 epochs of 10000 samples.

A.6.2 Fitting a mixture of Gaussians with Expectation-Maximisation

We also include an example where the model we fit is well-specified, which in this case means it is also a mixture of two Gaussians (Fig. S3. As directly optimizing a mixture distribution with gradient descent is not straightforward, we used an Expectation-Maximization algorithm (where we use the distances instead of the log-likelihood in the maximisation step)

Our model is specified by

$$p(x) = w_1\mathcal{N}(\mu_1, \Sigma_1) + w_2\mathcal{N}(\mu_2, \Sigma_2)$$

which we can write as a latent-variable model, where the latent variables are the cluster assignments:

$$p(x) = \sum_{k=1}^{2} p(x|z = k)p(z = k),$$

with $p(x|z = k) = \mathcal{N}(\mu_k, \Sigma_k)$ and $p(z = k) = w_k$.

We then iteratively performed the following two steps to optimise the model.

E-step: For each of the N datapoints $\hat{x}_i$ from p_{true}, we calculated the probability of it belonging to mixture component 1 or 2:

$$p(z = k|\hat{x}_i) = \frac{p(\hat{x}_i|z = k)p(z = k)}{\sum_{k'}^{2} p(\hat{x}_i|z = k')p(z = k')}.$$

M-step: We updated the mixture weights according to:

$$w_k = \frac{1}{N}\sum_{i}^{N} p(z = k|\hat{x}_i)$$

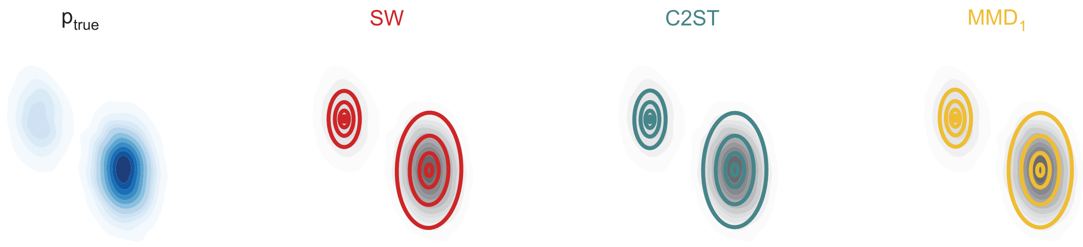

Figure S3: **Optimising distances in a well-specified model setting** When fitting a well-specified model (here, a mixture of two Gaussians), by using different distances in the loss, we can see that each model converges to the global optimum. Plotted are the contour lines of .25, .75, 1, and 2.5 standard deviations of the fitted Gaussians multiplied by their corresponding mixture weight.

Next, for each of the N datapoints, we first sampled a cluster assignment according to $z_i \sim p(z|\hat{x}_i)$. Then for each group of N_k datapoints assigned to cluster k we sampled N_k times according to $x_i \sim p(x|z_i)$, again using the reparameterisation trick. As before we computed the loss using a statistical distance, now separately for the two groups of samples assigned to either mixture component, and used gradient to optimise μ_k, Σ_k

Again, we used the ADAM optimizer (Kingma & Ba, 2015), with learning rate=0.01 and default momentums, using 2000 epochs of 5000 samples.

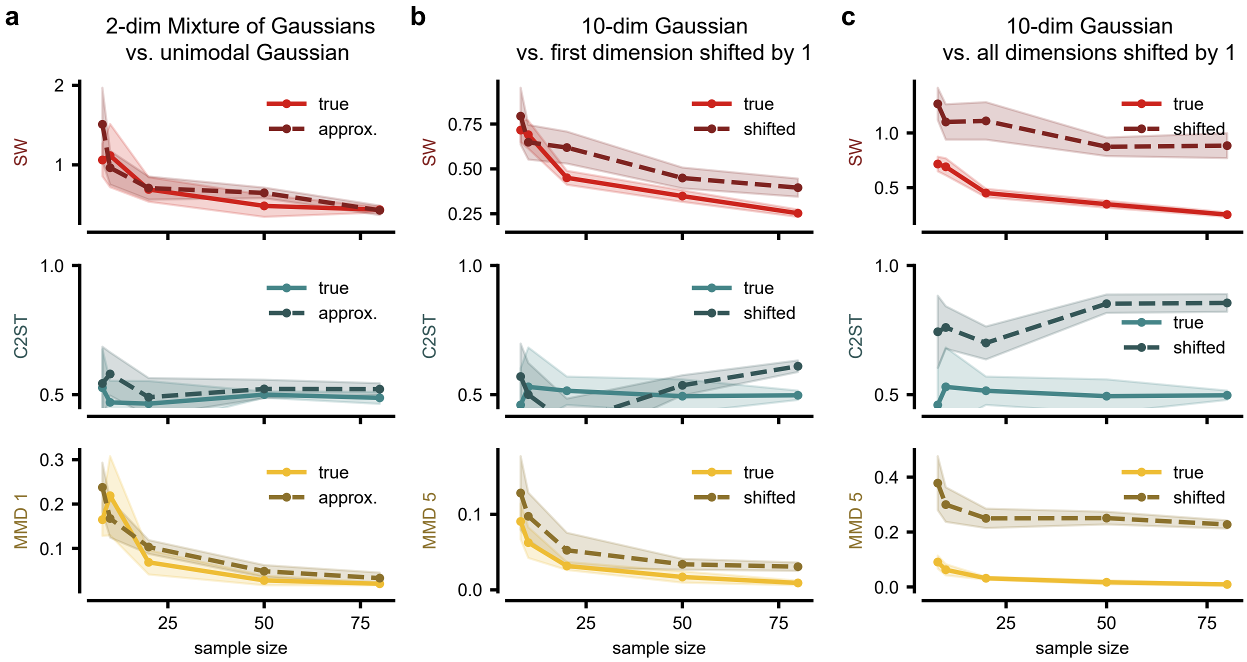

Figure S4: **The larger the difference between two distributions the fewer samples suffice to tell the true and shifted distribution apart.** We compare sample sets with varying sample sizes (between 8 and 80 samples per set) of a 'true' distribution either with a second dataset of the same distribution or with a sample set from an approximated/shifted distribution. We show the mean and standard deviation over five runs of randomly sampled data. **(a)** Distances for the 2d-MoG example shown in Fig. 1 compared to samples from a unimodal Gaussian approximation with the same mean and covariance. **(b)** Distances for a ten-dimensional standard normal distribution, for which *the first* dimension is shifted by one for the shifted example. **(c)** Distances for a ten-dimensional standard normal distribution, for which *all* dimensions are shifted by one for the shifted example.

A.7 Additional scaling experiments with different sample size budgets and ranges

In Fig. 8, we evaluated the robustness of the measures against the number of samples and the dimensionality of the data. We observed notably poor performance of the measures in scenarios with limited data. Here, we further examine the performance of the distances across datasets of varying sample sizes, particularly for small sample set sizes, ranging from only 8 to 80 samples per set (Fig. S4). We examine three distinct data configurations where the distinction between the true and approximated distributions progressively increases from subpanels S4 a to c. Across all distances, it becomes evident that the larger the disparity between the two distributions, the fewer samples are needed for differentiation. In the experiment where all dimensions are mean-shifted by one, a sample size of 8 is sufficient to distinguish between the distributions. However, for less distinct distributions, such as the unimodal Gaussian or a mean-shift by one in only one dimension (Supp. Fig. S4 a, b), all distances exhibit poor performance in distinguishing between the distributions.

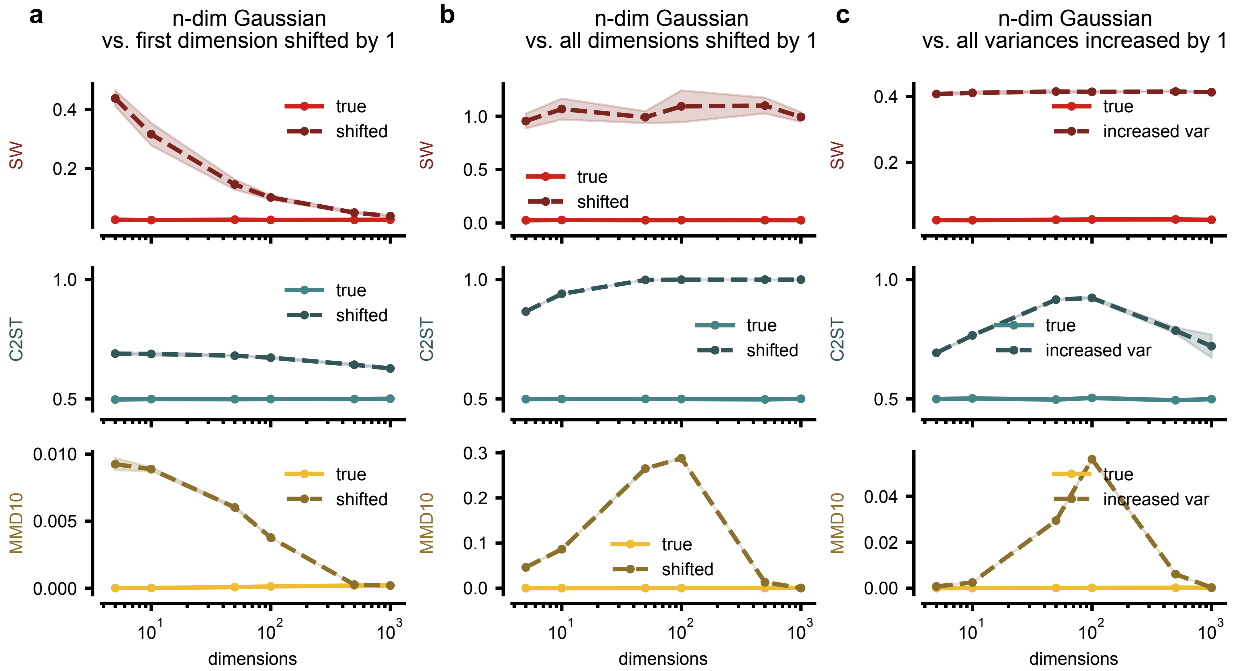

Figure S5: **The impact of dimensionality varies across distances, with certain distances facing particular challenges in higher dimensional spaces.** We compare sample sets of varying dimensionality (between 5 and 1000) of a ’‘true’ distribution either with a second dataset of the same distribution or with a sample set from an approximated/shifted distribution. The sample size is fixed to 10k for all experiments and we show the mean and standard deviation over five runs of randomly sampled data. The bandwidth parameter in Gaussian Kernel MMD is set to 10 for all experiments. **(a)** Distances for a sample set from an n-dimensional standard normal distribution, for which *the first* dimension is shifted by one. **(b)** Distances for a sample set from an n-dimensional standard normal distribution, for which *all* dimensions are shifted by one. **(c)** Distances for a sample set from an n-dimensional standard normal distribution, for which *variances are increased* by one for all dimensions.

A.8 Additional scaling experiments for different dimensionality of the data

When comparing the robustness of the measures with respect to the dimensionality of the data in Fig. 8, we observed a degradation in the ability to distinguish between distributions as dimensionalities increased. Notably, only the C2ST measure retained the capability to distinguish between the two distributions in higher dimensions which is aligned with the intuition that a classifier can easily pick up on differences in a single dimension. Extending this analysis, Fig. S5 presents similar experiments conducted on datasets where we compare an n-dimensional standard normal distribution with one where either all dimensions are mean-shifted by one (thus aligning with the C2ST experiment in Fig. 5b) or where all variances are increased by one. Fig. S5a corresponds to the experiment outlined in Fig. 8c on dimensionality. The bandwidth parameter for the MMD distance has been adjusted to suit the particular data configuration and is represented by the integer in the y-axis label. Generally, we notice that the Sliced-Wasserstein distance and MMD face difficulties in higher-dimensional spaces, especially when handling distributions that are only slightly distinct if the respective hyperparameters are kept constant across dimensions. In contrast, the C2ST distance consistently demonstrates good performance across all three experiments and for all ranges of dimensions.

A.9 Comparisons of practical compute times of different measures

Before computing such measures, in particular for scaling experiments such as the ones presented here, where measures are calculated across a large range of sample sizes N and data dimensions D, the practical compute time of the chosen measures should be considered. Depending on the downstream application, it might be time-critical to quickly evaluate distances which might favor some measures over others. Aligned with theoretical considerations regarding sampling complexity etc. as presented in Table 1), empirical compute times vary between the different measures. Given that empirical computational times for a single measure itself vary depending on the exact implementation, compute infrastructure, and problem at hand, we list approximated compute times for running the scaling experiments in Table S2. The calculated runtime combines both the comparisons of the 'true' and the 'shifted' or 'approximated' experiments. Each experiment contains five repeats across different sampled data subsets. The sample size experiment contains eight different sample size values N (50, 100, 200, 500, 1000, 2000, 3000, 4000), and the dimensionality experiment scales tests six different D (5, 10, 50, 100, 500, 1000). For more details, see Section 3. The version of C2ST we use here, which is based on NN-based classifiers, takes orders of magnitude longer to compute than SW and MMD. Note, however, that alternative implementations and classifier variants could speed this up.

	sample size experiment		**dimensionality experiment**
	2-dim MoGs	10-dim Gaussian	n-dim Gaussian
SWD	0.3 s	0.3 s	1.5 s
C2ST	150 s	300 s	1500 s
MMD	3 s	3 s	80 s

Table S2: CPU wallclock run times for the comparison scaling experiments in Fig.8. The runtime combines both the comparisons of the true vs. the shifted or approximated distribution. Values are rounded estimates.

A.10 Sensitivity of the MMD ~~bandwidth parameter~~kernel choices and hyperparameters

The general formulation of MMD allows for a wide range of kernel choices, each potentially with their own hyperparameters. These choices can have significant effect on its behavior. We provide some experiments demonstrating of the importance of well-tuned bandwidth parameters for Gaussian Kernel MMD as well as the impact of different kernels.

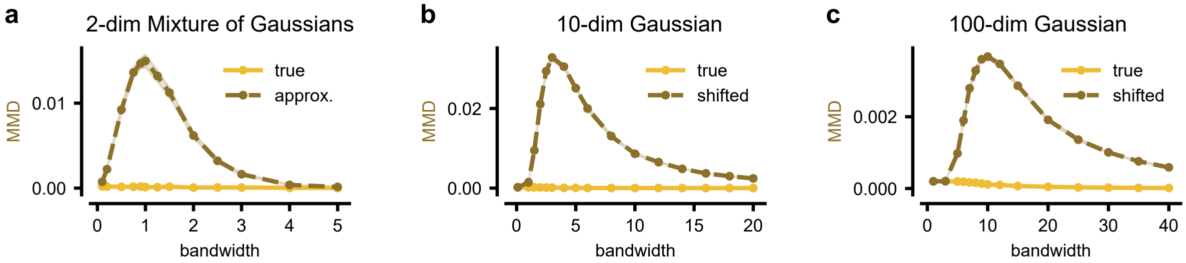

Figure S6: **The bandwidth parameter in Gaussian Kernel MMD is a sensitive parameter that requires careful selection for each dataset.** The sample size is fixed to 10k for all experiments and we show the mean and standard deviation over five runs of randomly sampled data. **(a)** ~~MMD^2~~ MMD^2 distance with varying bandwidth parameters between 0.1 and 5 for the 2d-MoG example compared to samples from a unimodal Gaussian approximation with the same mean and covariance. **(b)** ~~MMD^2~~ MMD^2 distance with varying bandwidth parameters between 0.1 and 20 for a 10-dimensional standard normal distribution, for which *the first* dimension is shifted by one for the shifted example. **(c)** ~~MMD^2~~ MMD^2 distance with varying bandwidth parameters between 1 and 40 for a 100-dimensional standard normal distribution, for which *the first* dimension is shifted by one for the shifted example.

We first vary the bandwidth parameter with fixed sample sizes for the three example datasets used in Section 3 (Fig. S6). We show that the estimated MMD values vary significantly across bandwidths, and both setting the bandwidth too low or too high yield poor results. However, we note that the values yielded by the median heuristic (bandwidths of 1, 5, and 10 for the three datasets, respectively, as shown in the main text) are quite near the peaks of the curves at which MMD most effectively distinguishes the distributions.

We then choose a set of bandwidth parameters to compare across the scaling experiments of Section 3 (Fig. S7). Again, poor choices of bandwidth values give misleading results, but bandwidth choices guided by the median heuristic generally perform well.

Finally, we vary the kernel choice across the scaling experiments of Section 3 (Fig. S7), using both a linear kernel (MMD_{lin}) and the distance-induced kernel corresponding to the standard energy distance (MMD_{en}). As expected, the linear kernel fails to distinguish distributions with matching means (2d-MoG) but performs reasonably well for distributions with mean-offsets even at high dimensions. The energy kernel performs similarly to the Gaussian kernel, without the added dependence on sensitive hyperparameters.

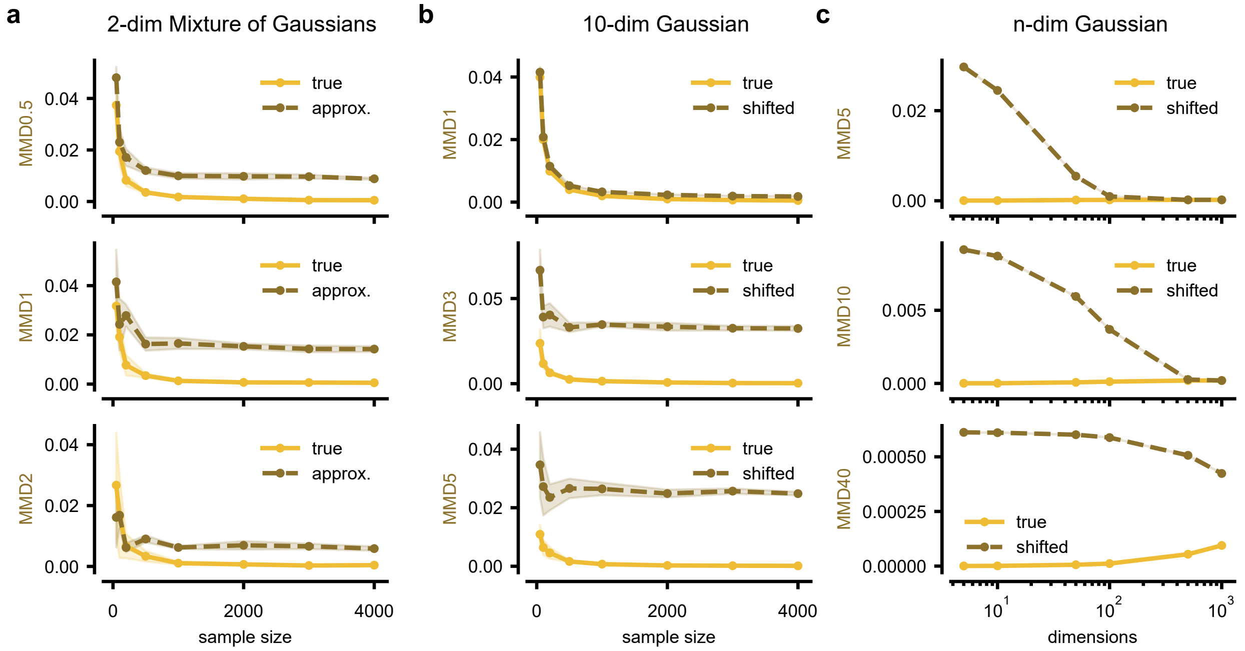

Figure S7: **Evaluating the effect of varying bandwidth parameters in Gaussian Kernel MMD for different sample sizes and dataset dimensionalities. (a,b)** We compare sample sets with varying sample sizes (between 50 and 4k samples per set) of a 'true' distribution either with a second dataset of the same distribution or with a sample set from an approximated/shifted distribution. We show the mean and standard deviation over five runs of randomly sampled data. **(a)** ~~MMD^2~~ MMD^2 distance with varying bandwidth parameters (0.5, 1, 2) for the 2d-MoG example shown in Fig. 1 compared to samples from a unimodal Gaussian approximation with the same mean and covariance. **(b)** ~~MMD^2~~ MMD^2 distance with varying bandwidth parameters (1, 3, 5) for a ten dimensional standard normal distribution, for which *the first* dimension is shifted by one for the shifted example. **(c)** ~~MMD^2~~ MMD^2 distance with varying bandwidth parameters (5, 10, 40) based on 10k samples from a standard normal distributions with varying dimensions (between 5 and 1000). As in (b) *the first* dimension is shifted by one for the 'shifted' dataset. Here we show one run due to computational costs.

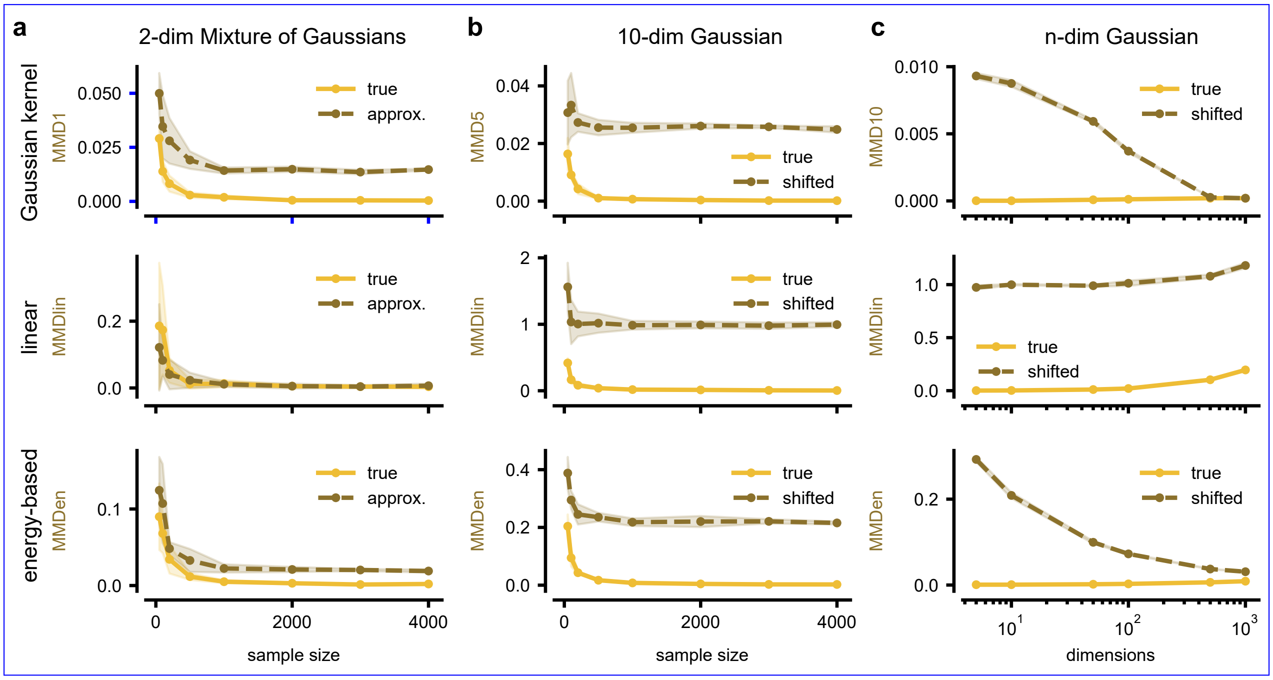

Figure S8: **Comparison of Gaussian Kernel MMD to different MMD kernels without tunable parameters.** We compare the performance of MMD^2 as presented in Fig.8 with a Gaussian kernel with bandwidth parameters adjusted for each dataset (1,5,10) (top row) to an MMD implementation with a linear kernel (middle row, MMD_{lin}; $k(x,y) = \langle x, y \rangle$) and an energy-distance based kernel, i.e., the kernel induced by the euclidean distance Sejdinovic et al. (2013) (bottom row, MMD_{en}; $k(x,y) = \|x\|^p + \|x\|^p - \|x-y\|^p$, with $p = 2$). The experiments, parameters for the Gaussian kernel bandwidth (indicated in the y-labels), and sample sizes etc. are identical to Fig. 8.

A.11 Additional results for ImageNet generative models

We generated 50,000 images using an unconditional diffusion model explicitly trained on ImageNet 64x64, as well as 100,000 class-conditional images from a conditional variant of the same model (Dockhorn et al., 2022) (we refer to it as GENIE). For additional comparison, we also generated 50,000 images using a consistency model explicitly trained on ImageNet 64x64 (Song et al., 2023) (we refer to it as CM). We compare these generated images to the ImageNet 64x64 test set.

For further comparison, we also evaluated image-generative models not specifically trained to reproduce ImageNet but designed for general-purpose image generation, such as Stable Diffusion and Midjourney (Rombach et al., 2022b; Midjourney, 2022). While these models might generate images that are more appealing to human observers, they do not necessarily produce images that align with the images contained in the ImageNet test set. We use the recently published million-scale dataset GenImage (Zhu et al., 2023). Especially, we include the models BigGAN Brock et al. (2018), ablated diffusion model (ADM) Dhariwal & Nichol (2021), Glide (Nichol et al., 2021), Vector Quantized Diffusion Model (VQDM) Gu et al. (2022), Wukong Wukong (2022), Stable diffusion 1.5 (SD1.5) Rombach et al. (2022b) and Midjourney Midjourney (2022).

It's important to note that not all models are specifically trained to capture ImageNet 64x64 images. For instance, models like Stable Diffusion and Midjourney are trained on much larger datasets (Schuhmann et al., 2022; Lin et al., 2014). Additionally, most of the models mentioned, except for ADM and BigGAN, are text-to-image generative models. To minimize significant distribution shifts, these models were prompted with the phrase "photo of [ImageNet class]" Zhu et al. (2023). Furthermore, all the other models generate images of larger resolution, which we resized to 64x64. Hence, we also only compare low-resolution features of natural images.

We evaluate each of the metrics on three random subsets, each consisting of 40,000 image embeddings. We show the average value for each model and metric in Table S3. Interestingly, both what is considered "closest" to ImageNet, as well as the relative ranking differs for different metrics, although with some consistent trends. Overall, the most recent unconditional models trained on ImageNet 64x64 perform best, as expected. However, which one is considered best differs for different metrics. Metrics that consider similar features of the distribution i.e., FID and MMD$_{\text{poly}}$ (statistics up to order 2 or 3) prefer GENIE, whereas universally consistent metrics do prefer CM (SWD and MMD$_{64}$). The estimated C2ST values differ based on the chosen classifier.

Overall, this analysis highlights that the choice of metric (i.e., what features it compares) and the specific implementation details (such as the classifier in C2ST estimates) matter and can lead to varying results.

Table S3: **Evaluating discrepancy to the ImageNet test set.**Each row presents various metrics computed on the Inception v3 embeddings of images. Columns correspond to different generative models. The initial three models are trained on ImageNet 64x64, serving as the reference point for comparison. Subsequent models are trained on alternative datasets or higher-resolution versions. In bold, we highlight the lowest value for each section of the table.

	GENIE	**CM**	**BigGAN**	**ADM**	**Glide**	**VQDM**	**WK**	**SD1.5**	**Midjourney**
FID	$\mathbf{5.1 \cdot 10^{0}}$	$5.3 \cdot 10^{0}$	$1.2 \cdot 10^{1}$	$1.1 \cdot 10^{1}$	$1.1 \cdot 10^{1}$	$\mathbf{9.8 \cdot 10^{0}}$	$1.1 \cdot 10^{1}$	$1.2 \cdot 10^{1}$	$1.0 \cdot 10^{1}$
SWD	$2.3 \cdot 10^{-2}$	$\mathbf{2.2 \cdot 10^{-2}}$	$5.3 \cdot 10^{-2}$	$5.1 \cdot 10^{-2}$	$4.9 \cdot 10^{-2}$	$\mathbf{4.0 \cdot 10^{-2}}$	$4.8 \cdot 10^{-2}$	$5.0 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$
MMD$_{64}$	$7.0 \cdot 10^{-5}$	$\mathbf{6.3 \cdot 10^{-5}}$	$1.9 \cdot 10^{-4}$	$1.9 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$	$\mathbf{1.5 \cdot 10^{-4}}$	$1.9 \cdot 10^{-4}$	$1.9 \cdot 10^{-4}$	$1.7 \cdot 10^{-4}$
MMD$_{\text{lin}}$	$2.5 \cdot 10^{-1}$	$\mathbf{2.3 \cdot 10^{-1}}$	$6.3 \cdot 10^{-1}$	$6.3 \cdot 10^{-1}$	$6.2 \cdot 10^{-1}$	$\mathbf{5.2 \cdot 10^{-1}}$	$6.2 \cdot 10^{-1}$	$6.4 \cdot 10^{-1}$	$6.1 \cdot 10^{-1}$
MMD$_{\text{poly}}$	$\mathbf{1.1 \cdot 10^{4}}$	$1.6 \cdot 10^{4}$	$3.1 \cdot 10^{4}$	$3.4 \cdot 10^{4}$	$3.2 \cdot 10^{4}$	$\mathbf{2.2 \cdot 10^{4}}$	$3.7 \cdot 10^{4}$	$3.1 \cdot 10^{4}$	$2.9 \cdot 10^{4}$
C2ST$_{\text{knn}}$	0.70	**0.69**	**0.81**	0.82	0.82	0.82	0.83	0.83	0.82
C2ST$_{\text{nn}}$	**0.72**	0.77	0.87	0.85	0.85	**0.85**	0.86	0.86	0.95

A.12 Details about scientific application examples

For the motion discrimination task, we used the decision times of a single animal during both correct and erroneous trials with dot motion coherence of 12.8%, leading to a one-dimensional dataset of ~~587 samples.~~ 1023 samples. From these 80% were used as a train set and 20% as a test set DDMs were implemented using the *pyDDM* toolbox (Shinn et al., 2020). DDM1 used a linear drift and exponential decision boundaries. In contrast, DDM2 used a constant drift and a constant decision boundary. Both were sampled 1,000 times to create the two synthetic datasets. The real chest X-ray dataset consists of 70,153 train samples and 25,596 test samples, the generated datasets from PGGAN and SD consist of 10,000 and 2,352 samples respectively.

In both applications we computed metrics between pairs of 10 random subsets from the compared distributions (scatter points on the violin plots). We computed the MMD with a bandwidth of 50 for the medical imaging datasets and a bandwidth of 0.5 for the decision time dataset.

References for Table S1

[1] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 2018.

[2] Mohamed Marouf, Pierre Machart, Vikas Bansal, Christoph Kilian, Daniel S Magruder, Christian F Krebs, and Stefan Bonn. Realistic in silico generation and augmentation of single-cell rna-seq data using generative adversarial networks. *Nature communications*, 2020.

[3] Snehalika Lall, Sumanta Ray, and Sanghamitra Bandyopadhyay. LSH-GAN enables in-silico generation of cells for small sample high dimensional scRNA-seq data. *Communications Biology*, 2022.

[4] Yazdan Zinati, Abdulrahman Takiddeen, and Amin Emad. GRouNdGAN: GRN-guided simulation of single-cell RNA-seq data using causal generative adversarial networks. *bioRxiv*, 2023.

[5] Wenzhuo Tang, Renming Liu, Hongzhi Wen, Xinnan Dai, Jiayuan Ding, Hang Li, Wenqi Fan, Yuying Xie, and Jiliang Tang. A general single-cell analysis framework via conditional diffusion generative models. *bioRxiv*, 2023.

[6] Robrecht Cannoodt, Wouter Saelens, Louise Deconinck, and Yvan Saeys. Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nature Communications*, 2021.

[7] Ofir Lindenbaum, Jay Stanley, Guy Wolf, and Smita Krishnaswamy. Geometry based data generation. *Advances in Neural Information Processing Systems*, 2018.

[8] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell RNA sequencing data. *Genome biology*, 2017.

[9] Helena L Crowell, Charlotte Soneson, Pierre-Luc Germain, Daniela Calini, Ludovic Collin, Catarina Raposo, Dheeraj Malhotra, and Mark D Robinson. Muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nature communications*, 2020.

[10] Wei Vivian Li and Jingyi Jessica Li. A statistical simulator scdesign for rational scRNA-seq experimental design. *Bioinformatics*, 2019.

[11] Tianyi Sun, Dongyuan Song, Wei Vivian Li, and Jingyi Jessica Li. scDesign2: a transparent simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured. *Genome biology*, 2021.

[12] Namid R Stillman and Roberto Mayor. Generative models of morphogenesis in developmental biology. In *Seminars in Cell & Developmental Biology*, 2023.

[13] Dominik JE Waibel, Ernst Röell, Bastian Rieck, Raja Giryes, and Carsten Marr. A diffusion model predicts 3d shapes from 2d microscopy images. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, 2023.

[14] Assaf Zaritsky, Andrew R Jamieson, Erik S Welf, Andres Nevarez, Justin Cillay, Ugur Eskiocak, Brandi L Cantarel, and Gaudenz Danuser. Interpretable deep learning uncovers cellular properties in label-free live cell images that are predictive of highly metastatic melanoma. *Cell systems*, 2021.

[15] Christopher J Soelistyo, Giulia Vallardi, Guillaume Charras, and Alan R Lowe. Learning biophysical determinants of cell fate with deep neural networks. *Nature Machine Intelligence*, 2022.

[16] Robert L Satcher and C Forbes Dewey. Theoretical estimates of mechanical properties of the endothelial cell cytoskeleton. *Biophysical journal*, 1996.

[17] Dimitrije Stamenović, Jeffrey J Fredberg, Ning Wang, James P Butler, and Donald E Ingber. A microstructural approach to cytoskeletal mechanics based on tensegrity. *Journal of Theoretical Biology*, 1996.

[18] D Stamenović and Donald E Ingber. Models of cytoskeletal mechanics of adherent cells. *Biomechanics and modeling in mechanobiology*, 2002.

[19] Mark F Coughlin and Dimitrije Stamenović. A prestressed cable network model of the adherent cell cytoskeleton. *Biophysical journal*, 2003.

[20] Guillaume Jouvet, Guillaume Cordonnier, Byungsoo Kim, Martin Lüthi, Andreas Vieli, and Andy Aschwanden. Deep learning speeds up ice flow modelling by several orders of magnitude. *Journal of Glaciology*, 2022.

[21] Guillaume Jouvet. Inversion of a stokes glacier flow model emulated by deep learning. *Journal of Glaciology*, 2023.

[22] Guillaume Jouvet and Guillaume Cordonnier. Ice-flow model emulator based on physics-informed deep learning. *Journal of Glaciology*, 2023.

[23] J. Bolibar, F. Sapienza, F. Maussion, R. Lguensat, B. Wouters, and F. Pérez. Universal differential equations for glacier ice flow modelling. *Geoscientific Model Development*, 2023.

[24] Vincent Verjans and Alexander Robel. Accelerating subglacial hydrology for ice sheet models with deep learning methods. *Geophysical Research Letters*, 2024.

[25] R. Winkelmann, M. A. Martin, M. Haseloff, T. Albrecht, E. Bueler, C. Khroulev, and A. Levermann. The potsdam parallel ice sheet model (PISM-PIK) – Part 1: Model description. *The Cryosphere*, 2011.

[26] E. Larour, H. Seroussi, M. Morlighem, and E. Rignot. Continental scale, high order, high spatial resolution, ice sheet modeling using the ice sheet system model (ISSM). *Journal of Geophysical Research: Earth Surface*, 2012.

[27] O. Gagliardini, T. Zwinger, F. Gillet-Chaulet, G. Durand, L. Favier, B. de Fleurian, R. Greve, M. Malinen, C. Martín, P. Råback, J. Ruokolainen, M. Sacchettini, M. Schäfer, H. Seddik, and J. Thies. Capabilities and performance of elmer/ice, a new-generation ice sheet model. *Geoscientific Model Development*, 2013.

[28] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 2023.

[29] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: Learning a generative model from a single natural image. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[30] David John Gagne II, Sue Ellen Haupt, Douglas W. Nychka, and Gregory Thompson. Interpretable deep learning for spatial analysis of severe hailstorms. *Monthly Weather Review*, 2019.

[31] Jean Côté, Sylvie Gravel, André Méthot, Alain Patoine, Michel Roch, and Andrew Staniforth. The operational CMC–MRB global environmental multiscale (GEM) model. Part i: Design considerations and formulation. *Monthly Weather Review*, 1998.

[32] Linjiong Zhou, Shian-Jiann Lin, Jan-Huey Chen, Lucas M. Harris, Xi Chen, and Shannon L. Rees. Toward convective-scale prediction within the next generation global prediction system. *Bulletin of the American Meteorological Society*, 2019.

[33] L. Magnusson, J.-R. Bidlot, M. Bonavita, A. R. Brown, P. A. Browne, G. De Chiara, M. Dahoui, S. T. K. Lang, T. McNally, K. S. Mogensen, F. Pappenberger, F. Prates, F. Rabier, D. S. Richardson, F. Vitart, and S. Malardel. ECMWF activities for improved hurricane forecasts. *Bulletin of the American Meteorological Society*, 2019.

[34] Dylan M Anstine and Olexandr Isayev. Generative models as an emerging paradigm in the chemical sciences. *Journal of the American Chemical Society*, 2023.

[35] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 2018.

[36] Robin Winter, Floriane Montanari, Andreas Steffen, Hans Briem, Frank Noé, and Djork-Arné Clevert. Efficient multi-objective molecular optimization in a continuous latent space. *Chemical science*, 2019.

[37] Jaechang Lim, Seongok Ryu, Jin Woo Kim, and Woo Youn Kim. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of cheminformatics*, 2018.

[38] Benjamin Sanchez-Lengeling, Carlos Outeiral, Gabriel L Guimaraes, and Alan Aspuru-Guzik. Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). *ChemRxiv*, 2017.

[39] Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

[40] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, 2022.

[41] Lei Huang, Hengtong Zhang, Tingyang Xu, and Ka-Chun Wong. Mdm: Molecular diffusion model for 3d molecule generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[42] Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. *Advances in Neural Information Processing Systems*, 2022.

[43] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 2018.

[44] Niclas Ståhl, Goran Falkman, Alexander Karlsson, Gunnar Mathiason, and Jonas Bostrom. Deep reinforcement learning for multiparameter optimization in de novo drug design. *Journal of chemical information and modeling*, 2019.

[45] Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Simon Blackburn, Karam Thomas, Connor Coley, Jian Tang, et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International conference on machine learning*, 2020.

[46] Jeffrey Regier, Andrew Miller, Jon McAuliffe, Ryan Adams, Matt Hoffman, Dustin Lang, David Schlegel, and Mr Prabhat. Celeste: Variational inference for a generative model of astronomical images. In *International Conference on Machine Learning*, 2015.

[47] François Lanusse, Rachel Mandelbaum, Siamak Ravanbakhsh, Chun-Liang Li, Peter Freeman, and Barnabás Póczos. Deep generative models for galaxy image simulations. *Monthly Notices of the Royal Astronomical Society*, 2021.

[48] Michael J Smith and James E Geach. Generative deep fields: arbitrarily sized, random synthetic astronomical images through deep learning. *Monthly Notices of the Royal Astronomical Society*, 2019.

[49] Rachel Mandelbaum, Christopher M Hirata, Alexie Leauthaud, Richard J Massey, and Jason Rhodes. Precision simulation of ground-based lensing data using observations from space. *Monthly Notices of the Royal Astronomical Society*, 2012.

[50] Barnaby TP Rowe, Mike Jarvis, Rachel Mandelbaum, Gary M Bernstein, James Bosch, Melanie Simet, Joshua E Meyers, Tomasz Kacprzak, Reiko Nakajima, Joe Zuntz, et al. GALSIM: the modular galaxy image simulation toolkit. *Astronomy and Computing*, 2015.