A Kernelised Stein Statistic for Assessing Implicit Generative Models

Wenkai Xu
Department of Statistics
University of Oxford
wenkai.xu@stats.ox.ac.uk

Gesine Reinert
Department of Statistics
University of Oxford
and Alan Turing Institute
reinert@stats.ox.ac.uk

Abstract

Synthetic data generation has become a key ingredient for training machine learning procedures, addressing tasks such as data augmentation, analysing privacy-sensitive data, or visualising representative samples. Assessing the quality of such synthetic data generators hence has to be addressed. As (deep) generative models for synthetic data often do not admit explicit probability distributions, classical statistical procedures for assessing model goodness-of-fit may not be applicable. In this paper, we propose a principled procedure to assess the quality of a synthetic data generator. The procedure is a kernelised Stein discrepancy (KSD)-type test which is based on a non-parametric Stein operator for the synthetic data generator of interest. This operator is estimated from samples which are obtained from the synthetic data generator and hence can be applied even when the model is only implicit. In contrast to classical testing, the sample size from the synthetic data generator can be as large as desired, while the size of the observed data which the generator aims to emulate is fixed. Experimental results on synthetic distributions and trained generative models on synthetic and real datasets illustrate that the method shows improved power performance compared to existing approaches.

1 Introduction

Synthetic data capturing main features of the original dataset are of particular interest for machine learning methods. The use of original dataset for machine learning tasks can be problematic or even prohibitive in certain scenarios, e.g. under authority regularisation on privacy-sensitive information, training models on small-sample dataset, or calibrating models with imbalanced groups. High quality synthetic data generation procedures surpass some of these challenges by creating de-identified data to preserve privacy and to augment small or imbalanced datasets. Training deep generative models has been widely studied in recent years [Kingma and Welling, 2013, Radford et al., 2015, Song and Kingma, 2021] and methods such as those based on Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] provide powerful approaches that learn to generate synthetic data which resemble the original data distributions. However, these deep generative models usually do not provide theoretical guarantees on the *goodness-of-fit* to the original data [Creswell et al., 2018].

To the best of our knowledge, existing mainstream developments for deep generative models [Song and Ermon, 2020, Li et al., 2017] do not provide a systematic approach to assess the quality of the synthetic samples. Instead, heuristic methods are applied; e.g. for image data, the quality of samples are generally decided via visual comparisons. The study of the training quality has been relying largely on the specific choice of training loss, which does not directly translate into a measure of sample quality; in the case of the log-likelihood see for example Theis et al. [2015]. Common quality assessment measures for implicit generative models, on images for example, include Inception Scores

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

(IS) [Salimans et al., 2016] and Fréchet Inception Distance (FID) [Heusel et al., 2017], which are motivated by human inception systems in the visual cortex and pooling [Wang et al., 2004]. Bińkowski et al. [2018] pointed out issues for IS and FID and developed the Kernel Inception Distance (KID) for more general datasets. Although these scores can be used for for comparisons, they do not provide a statistical significance test which would assess whether a deemed *good* generative model is "*good enough*". A key stumbling block is that the distribution from which a synthetic method generates samples is not available; one only ever observes samples from it.

For models in which the density is known explicitly, at least up to a normalising constant, some assessment methods are available. Gorham and Mackey [2015] proposed to assess sample quality using discrepancy measures based on Stein's method; Chwialkowski et al. [2016] and Liu et al. [2016] created a kernelised test statistic called *kernelised Stein discrepancy* (KSD) based on Stein operators and used the unit ball of an RKHS as test functions. Schrab et al. [2022] assesses the quality of generative models on the MNIST image dataset from LeCun et al. [1995] using an aggregated kernel Stein discrepancy (KSDAgg) test; still an explicit density is required. The only available implicit goodness-of-fit test, AgraSSt [Xu and Reinert, 2022], applies only to generators of finite graphs; it is also of KSD form and makes extensive use of the discrete and finite nature of the problem. To date, quality assessment procedures of *implicit* deep generative models for continuous data remains unresolved. This paper addresses this gap.

The underlying idea can be sketched as follows. Traditionally, given a set of n observations, each in $\mathbb{R}^m$, one would estimate the distribution of these observations from the data and then check whether the synthetic data can be viewed as coming from the data distribution. Here instead we characterise the distribution which is generated possibly implicitly from the synthetic data generator, and then test whether the observed data can be viewed as coming from the synthetic data distribution. The advantage of this approach is that while the observed sample size n may be fairly small, the synthetic data distribution can be estimated to any desirable level of accuracy by generating a large number of samples from it. Similarly to the works mentioned in the previous paragraph for goodness-of-fit tests, we use a KSD approach, based on a Stein operator which characterises the synthetic data distribution. As the synthetic data generator is usually implicit, this Stein operator is not available. We show however that under mild assumptions it can be estimated from synthetic data samples to any desired level of accuracy.

Our contributions We introduce a method to assess (deep) generative models, which are often *black-box* approaches, when the underlying probability distribution is continuous, usually in high-dimensions. To this purpose, we develop a non-parametric Stein operator and the corresponding non-parametric kernel Stein discrepancies (NP-KSD), based on estimating conditional score functions. Moreover, we give theoretical guarantees for NP-KSD.

This paper is structured as follows. We start with a review of Stein's method and KSD goodness-of-fit tests for explicit models in Section 2 before in Section 3 we introduce NP-KSD and analyse the model assessment procedures. We show results of experiments in Section 4 and conclude with future directions in Section 5. Theoretical underpinnings, and additional results are provided in the supplementary material. The code is available at `https://github.com/wenkaixl/npksd.git`.

2 Stein's method and kernel Stein discrepancy tests

Stein identities, equations, and operators Stein's method [Stein, 1972] provides an elegant tool to characterise distributions via *Stein operators*, which can be used to assess distances between probability distributions, see for example Barbour and Chen [2005], Mijoule et al. [2021]. Given a distribution q, an operator $\mathcal{A}_q$ is called a Stein operator w.r.t. q and *Stein class* $\mathcal{F}$ if the following Stein identity holds for any *test function* $f \in \mathcal{F}$: $\mathbb{E}_q[\mathcal{A}_q f] = 0$. For a test function h one then aims to find a function $f = f_h \in \mathcal{F}$ which solves the *Stein equation*

$$\mathcal{A}_q f(\boldsymbol{x}) = h(\boldsymbol{x}) - \mathbb{E}_q[h(\boldsymbol{x})]. \tag{1}$$

Then for any distribution p, taking expectations $\mathbb{E}_p$ in Eq. 1 assesses the distance $|\mathbb{E}_p h - \mathbb{E}_q h|$ through $|\mathbb{E}_p \mathcal{A}_q f|$, an expression in which randomness enters only through the distribution p.

When the density function q is given explicitly, with smooth support $\Omega_q \subset \mathbb{R}^m$, is differentiable and vanishes at the boundary of Ω_q, a common choice of Stein operator in the literature is based on the

score-function $\mathbf{s}_q = \nabla \log q$ (with the convention that $\mathbf{s}_q \equiv 0$ outside of Ω_q), see for example Mijoule et al. [2021]. Here, ∇ denotes the gradient operator and taken to be a column vector. The *score-Stein operator*[1] is the vector-valued operator acting on (vector-valued) function $\mathbf{f}$,

$$\mathcal{A}_q \mathbf{f}(\boldsymbol{x}) = \mathbf{f}(\boldsymbol{x})^\top \nabla \log q(\boldsymbol{x}) + \nabla \cdot \mathbf{f}(\boldsymbol{x}), \tag{2}$$

and the Stein identity $\mathbb{E}_q[\mathcal{A}_q f] = 0$ holds for functions f which belong to the so-called *canonical Stein class* defined in Mijoule et al. [2021], Definition 3.2. As it requires knowledge of the density q only via its score function, this Stein operator is particularly useful for unnormalised densities [Hyvärinen, 2005], appearing e.g. in energy based models (EBM) [LeCun et al., 2006].

Kernel Stein discrepancy To assess discrepancies between two probability distributions p and q, the Stein discrepancy (w.r.t. class $\mathcal{B} \subset \mathcal{F}$) is defined as [Gorham and Mackey, 2015]

$$\mathrm{SD}(p\|q, \mathcal{B}) = \sup_{f \in \mathcal{B}} \{|\mathbb{E}_p[\mathcal{A}_q f] - \underbrace{\mathbb{E}_p[\mathcal{A}_p f]}_{=0}|\} = \sup_{f \in \mathcal{B}} |\mathbb{E}_p[\mathcal{A}_q f]|. \tag{3}$$

As the $\sup f$ over a general class $\mathcal{B}$ can be difficult to compute, taking $\mathcal{B}$ as the unit ball of a reproducing kernel Hilbert space (RKHS) has been suggested in Chwialkowski et al. [2016], Liu et al. [2016], resulting in the *kernel Stein discrepancy* (KSD) defined as

$$\mathrm{KSD}(p\|q, \mathcal{H}) = \sup_{f \in \mathcal{B}_1(\mathcal{H})} |\mathbb{E}_p[\mathcal{A}_q f]|. \tag{4}$$

Denoting by k the reproducing kernel associated with the RKHS $\mathcal{H}$ over a set $\mathcal{X}$, the reproducing property ensures that $\forall f \in \mathcal{H}$, $f(\boldsymbol{x}) = \langle f, k(\boldsymbol{x}, \cdot)\rangle_{\mathcal{H}}, \forall \boldsymbol{x} \in \mathcal{X}$. Algebraic manipulations yield

$$\mathrm{KSD}^2(q\|p) = \mathbb{E}_{\boldsymbol{x}, \tilde{\boldsymbol{x}} \sim p}[u_q(\boldsymbol{x}, \tilde{\boldsymbol{x}})], \tag{5}$$

where $u_q(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \langle \mathcal{A}_q k(\boldsymbol{x}, \cdot), \mathcal{A}_q k(\tilde{\boldsymbol{x}}, \cdot)\rangle_{\mathcal{H}}$, which takes the exact $\sup$ without approximation and does not involve the (sample) distribution p. Then, KSD^2 can be estimated through empirical means, over samples from p. For example, V-statistics [Van der Vaart, 2000] and U-statistics [Lee, 1990] estimates are

$$\mathrm{KSD}_v^2(q\|p) = \frac{1}{m^2} \sum_{i,j} u_q(\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad \mathrm{KSD}_u^2(q\|p) = \frac{1}{m(m-1)} \sum_{i \neq j} u_q(\boldsymbol{x}_i\, \boldsymbol{x}_j). \tag{6}$$

KSD has been proposed as discrepancy measure between distributions for testing model goodness-of-fit in Chwialkowski et al. [2016] and Liu et al. [2016].

KSD testing procedure Suppose we have observed samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ from the *unknown* distribution p. To test the null hypothesis $H_0 : p = q$ against the (broad class of) alternative hypothesis $H_1 : p \neq q$, KSD can be empirically estimated via Eq. 6. The null distribution is usually simulated via the wild-bootstrap procedure [Chwialkowski et al., 2014]. Then if the empirical quantile, i.e. the proportion of wild bootstrap samples that are larger than $\mathrm{KSD}_v^2(q\|p)$, is smaller than the pre-defined test level (or significance level) α, the null hypothesis is rejected; otherwise the null hypothesis is not rejected. In this way, a systematic non-parametric goodness-of-fit testing procedure is obtained, which is applicable to unnormalised models.

3 Non-Parametric kernel Stein discrepancies

The construction of a KSD relies on the knowledge of the density model, up to normalisation. However, for deep generative models where the density function is not explicitly known, the computation of the Stein operator in Eq. 2, which is based on an explicit parametric density, is no longer feasible.

While in principle one could estimate the multivariate density function from synthetic data, density estimation in high dimensions is known to be problematic, see for example Scott and Sain [2005]. Instead, Stein's method allows to use a two-step approach: For data in $\mathbb{R}^m$, we first pick a coordinate $i \in [m] := \{1, \ldots, m\}$, and then we characterize the univariate conditional distribution of that coordinate, given the values of the other coordinates. Using score Stein operators from Ley et al. [2017], this approach only requires knowledge or estimation of univariate conditional score functions.

[1] also referred to as Langevin Stein operator [Barp et al., 2019].

First we introduce some notation. Observed data $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ are taken to be column vectors, $\boldsymbol{z}_i = (z_i^{(1)}, \ldots, z_i^{(m)})^\top \in \mathbb{R}^m$. For the generative model G we write $\boldsymbol{X} \sim G$ to denote a random $\mathbb{R}^m$-valued element from the (often only given implicitly) distribution which is underlying G. Using G, we generate N samples denoted by $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$. In our case, n is fixed and $n \ll N$, allowing $N \to \infty$ in theoretical results. The kernel of an RKHS is denoted by k and is assumed to be bounded. For $\boldsymbol{x} \in \mathbb{R}^m$, $x \in \mathbb{R}$ and $g(\boldsymbol{x}) : \mathbb{R}^m \to \mathbb{R}$, we write $g_{x^{(-i)}}(x) : \mathbb{R} \to \mathbb{R}$ for the univariate function which acts only on the coordinate i and fixes the other coordinates to equal $x^{(j)}, j \neq i$, so that $g_{x^{(-i)}}(x) = g(x^{(1)}, \ldots, x^{(i-1)}, x, x^{(i+1)}, \ldots, x^{(m)})$.

For $i \in [m]$ let $\mathcal{T}^{(i)}$ denote a Stein operator for the conditional distribution $Q^{(i)} = Q^{(i)}_{x^{(-i)}}$ with $\mathbb{E}_{Q^{(i)}_{x^{(-i)}}} g_{x^{(-i)}}(x) = \mathbb{E}[g_{y^{(-i)}}(Y) | Y^{(j)} = y^{(j)}, j \neq i]$. The proposed Stein operator $\mathcal{A}$ acting on functions $g : \mathbb{R}^m \to \mathbb{R}$ underlying the non-parametric Stein operator is

$$\mathcal{A}g(x^{(1)}, \ldots, x^{(m)}) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{T}^{(i)} g_{x^{(-i)}}(x^{(i)}). \tag{7}$$

We note that for $\boldsymbol{X} \sim q$, the Stein identity $\mathbb{E}\mathcal{A}g(\boldsymbol{X}) = 0$ holds and thus $\mathcal{A}$ is a Stein operator. The domain of the operator will depend on the conditional distribution in question. Instead of using the weights $w_i = \frac{1}{m}$ for the $\mathcal{T}^{(i)}$, other positive weights which sum to 1 would be possible, but for simplicity we use equal weights. A more detailed theoretical justification of Eq. 7 is given in Appendix A.

In what follows we use as Stein operator for a differentiable univariate density q the score operator from Eq. 2, given by

$$\mathcal{T}_q^{(i)} f(x) = f'(x) + f(x) \frac{q'(x)}{q(x)}. \tag{8}$$

In Proposition D.1 of Appendix D we shall see that the operator in Eq. 7 equals the score-Stein operator in Eq. 2; in Appendix D an example is also given. For the development in this paper, Eq. 7 is more convenient as it relates directly to conditional distributions. Other choices of Stein operators are discussed for example in Ley et al. [2017], Mijoule et al. [2021], Xu [2022].

Re-sampling Stein operators The Stein operator Eq. 7 depends on all coordinates $i \in [m]$. When m is large we can estimate this operator via re-sampling with replacement, as follows. We draw B samples $\{i_1, \ldots, i_B\}$ with replacement from $[m]$ such that $\{i_1, \ldots, i_B\} \sim \mathrm{Multinom}(B, \{\frac{1}{m}\}_{i \in [m]})$. The re-sampled Stein operator acting on $f : \mathbb{R}^m \to \mathbb{R}$ is

$$\mathcal{A}^B f(\boldsymbol{z}) := \frac{1}{B} \sum_{b=1}^{B} \mathcal{A}^{(i_b)} f(\boldsymbol{z}). \tag{9}$$

Then we have $\mathbb{E}\mathcal{A}^B f(\boldsymbol{X}) = \frac{1}{B} \sum_{b=1}^{B} \mathbb{E}\mathcal{A}^{(i_b)} f(\boldsymbol{X}) = 0$. So $\mathcal{A}^B$ is again a Stein operator.

In practice, when m is large, the stochastic operator in Eq. 9 creates a computationally efficient way for comparing distributions. A similar re-sampling strategy for constructing stochastic operators has been suggested in the context of Bayesian inference [Gorham et al., 2020], where conditional score functions, which are given in parametric form, are re-sampled to derive score-based (or Langevin) Stein operators for posterior distributions. The conditional distribution in Stein operators has been considered for example in Singhal et al. [2019] and Wang et al. [2018]; see also Zhuo et al. [2018] and Liu and Wang [2016] in the context of graphical models. In graphical models, the conditional distribution is simplified to conditioning on the Markov blanket [Wang et al., 2018], which is a subset of the full coordinate; however, no random re-sampling is used. Conditional distributions also apply in message passing, but there, the sequence of updates is ordered.

Estimating Stein operators via score matching Usually the score function q'/q in Eq. 8 is not available but needs to be estimated. An efficient way of estimating the score function is through score-matching, see for example Hyvärinen [2005], Song and Kingma [2021], Wenliang et al. [2019]. Score matching relies on the following score-matching (SM) objective [Hyvärinen, 2005],

$$J(p \| q) = \mathbb{E}_p \left[\| \nabla \log p(\boldsymbol{x}) - \nabla \log q(\boldsymbol{x}) \|^2 \right], \tag{10}$$

Algorithm 1 Estimating the conditional probability via summary statistics

Input: Generator G; summary statistics $t(\cdot)$; number of samples N from G; re-sample size B
Procedure:
1: Generate samples $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ from G.
2: Generate coordinate index sample $\{i_1, \ldots, i_B\}$
3: For $i_b \in [m], l \in [N]$, estimate $q(z^{(i_b)}|t(z^{-i_b})$ from samples $\{y_l^{(i_b)}, t(y_l{}^{-i_b})\}_{l\in[N]}$ via the score-matching objective in Eq. 10.
Output: $\widehat{s}_{t,N}^{(i)}(z^{(i)}|t(z^{(-i)})), \forall i \in [m]$.

which is particularly useful for unnormalised models such as EBMs. Additional details are included in Appendix E. Often score matching estimators can be shown to be consistent, see for example Song et al. [2020]. Proposition 3.1, proven in Appendix B, gives theoretical guarantees for the consistency of a general form of Stein operator estimation, as follows.

Proposition 3.1. *Suppose that for $i \in [m]$, $\widehat{s}_N^{(i)}$ is a consistent estimator of the univariate score function $s^{(i)}$. Let $\mathcal{T}^{(i)}$ be a Stein operator for the univariate differentiable probability distribution $Q^{(i)}$ of the generalised density operator form Eq. 8. Let*

$$\widehat{\mathcal{T}}_N^{(i)} g(x) = g'(x) + g(x)\widehat{s}_N^{(i)} \qquad \text{and} \qquad \widehat{\mathcal{A}} g = \frac{1}{m}\sum_i \widehat{\mathcal{T}}_N^{(i)} g_{x^{(-i)}}.$$

Then $\widehat{\mathcal{T}}_N^{(i)}$ is a consistent estimator for $\mathcal{T}^{(i)}$, and $\widehat{\mathcal{A}}$ is a consistent estimator of $\mathcal{A}$.

Non-parametric Stein operators with summary statistics In practice, the data $y^{(-i)} \in \mathbb{R}^{m-1}$ can be high dimensional, e.g. image pixels, and the observations can be sparse. Thus, estimation of the conditional distribution can be unstable or exponentially large sample size is required. Motivated by Xu and Reinert [2021] and Xu and Reinert [2022], we use a low-dimensional measurable non-trivial summary statistic t and the conditional distribution of the data given t as new target distributions. Heuristically, if two distributions match, then so do their conditional distributions. Thus, the conditional distribution $Q^{(i)}(A)$ is replaced by $Q_t^{(i)}(A) = \mathbb{P}(X^{(i)} \in A|t(x^{(-i)}))$. Setting $t(x^{(-i)}) = x^{(-i)}$ replicates the actual conditional distribution. We denote the univariate score function of $q_t(x|t(x^{(-i)}))$ by $s_t^{(i)}(x|t(x^{(-i)}))$, or by $s_t^{(i)}(x)$ when the context is clear. The summary statistics $t(x^{(-i)})$ can be univariate or multivariate, and they may attempt to capture useful distributional features. Here we consider univariate summary statistics such as the sample mean.

Taking these component-dependent conditional Stein operators results in the Stein operator in general to be no longer equal to the Langevin Stein operator of the corresponding conditional multivariate distribution. Thus, Proposition D.1 (presented in the Appendix) no longer applies.

The non-parametric Stein operator enables the construction of Stein-based statistics based on Eq. 7with estimated score functions $\widehat{s}_{t,N}^{(i)}$ using generated samples from the model G, as shown in Algorithm 1. The re-sampled non-parametric Stein operator is

$$\widehat{\mathcal{A}_{t,N}^B} g = \frac{1}{B}\sum_b \widehat{\mathcal{T}}_{t,N}^{(i_b)} g_{x^{(-i_b)}} = \frac{1}{B}\sum_b \Big(g'_{x^{(-i_b)}} + g_{x^{(-i_b)}}\widehat{s}_{t,N}^{(i)}\Big). \tag{11}$$

Non-parametric kernel Stein discrepancy Using the non-parametric Stein operator in Eq. 11 we define the corresponding non-parametric Stein discrepancy (NP-KSD) employing the Stein discrepancy notion in Eq. 3 and choosing as set of test functions the unit ball of the RKHS. Similarly to Eq. 4, we define the NP-KSD with summary statistic t as

$$\text{NP-KSD}_t(G\|p) = \sup_{f\in\mathcal{B}_1(\mathcal{H})} \mathbb{E}_p[\widehat{\mathcal{A}}_{t,N}^B f]. \tag{12}$$

A similar quadratic form as in Eq. 5 applies to give

$$\text{NP-KSD}_t^2(G\|p) = \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\widehat{u}_{t,N}^B(\boldsymbol{x},\tilde{\boldsymbol{x}})], \tag{13}$$

where $\widehat{u}^B_{t,N}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \langle \widehat{\mathcal{A}}^B_{t,N} k(\boldsymbol{x}, \cdot), \widehat{\mathcal{A}}^B_{t,N} k(\tilde{\boldsymbol{x}}, \cdot) \rangle_{\mathcal{H}}$. The empirical estimate is

$$\widehat{\text{NP-KSD}}^2_t(G\|p) = \frac{1}{n^2} \sum_{i,j \in [n]} [\widehat{u}^B_{t,N}(\boldsymbol{z}_i, \boldsymbol{z}_j)], \tag{14}$$

where $\mathbb{S} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\} \sim p$. Thus, NP-KSD allows the computation between a set of samples and a generative model, enabling the quality assessment of synthetic data generators for implicit models.

The relationship between NP-KSD and KSD is clarified in the following result; we use the notation $\hat{\mathbf{s}}_{t,N} = (\hat{s}_{t,N}(x^{(i)}), i \in [m])$. Here we set

$$\text{KSD}^2_t(q_t\|p) = \mathbb{E}_{\boldsymbol{x}, \tilde{\boldsymbol{x}} \sim p}[\langle \mathcal{A}_t k(\boldsymbol{x}, \cdot), \mathcal{A}_t k(\tilde{\boldsymbol{x}}, \cdot) \rangle_{\mathcal{H}} \quad \text{with} \quad \mathcal{A}_t g(\boldsymbol{x}) := \frac{1}{m} \sum_{i=1}^{m} \mathcal{T}^{(i)}_{q_t} g_{x^{(-i)}}(x^{(i)}) \tag{15}$$

as in Eq. 7, and following Eq. 8, $\mathcal{T}^{(i)}_{q_t} g_{x^{(-i)}}(x) = g'_{x^{(-i)}}(x) + g_{x^{(-i)}}(x) s^{(i)}_t(x|t(x^{(-i)}))$. More details about the interpretation of this quantity are given in Appendix B.1.

Theorem 3.2. *Assume that the score function estimator vector* $\hat{\mathbf{s}}_{t,N} = (\hat{s}^{(i)}_{t,N}, i = 1, \ldots, m)^\top$ *is asymptotically normal with mean* 0 *and covariance matrix* $N^{-1}\Sigma_s$*. Then* $\text{NP-KSD}^2_t(G\|p)$ *converges in probability to* $\text{KSD}^2_t(q_t\|p)$ *at rate at least* $\min(B^{-\frac{1}{2}}, N^{-\frac{1}{2}})$.

The proof of Theorem 3.2, which is found in Appendix B, also shows that the distribution $\text{NP-KSD}^2_t(G\|p) - \text{KSD}^2_t(q_t\|p)$ involves a mixture of normal variables. The assumption of asymptotic normality for score matching estimators is often satisfied, see for example Song et al. [2020].

Model assessment with NP-KSD Given an implicit generative model G and a set of observed samples $\mathbb{S} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$, we aim to assess whether the observed samples $\mathbb{S}$ are consistent with the hypothesis that they were generated from G. This test assumes that samples generated from G follow some (unknown) distribution q and that the samples $\mathbb{S}$ are generated according to some (unknown) distribution p. The null hypothesis is $H_0 : p = q$ while the alternative is $H_1 : p \neq q$. We note that the observed sample size n is fixed.

NP-KSD testing procedures NP-KSD can be applied for testing the above hypothesis using the testing procedure outlined in Algorithm 2. In contrast to the KSD testing procedure in Section 2, the NP-KSD test in Algorithm 2 is a Monte Carlo based test [Xu and Reinert, 2021, 2022, Schrab et al., 2022] for which the null distribution is approximated via samples generated from G instead of the wild bootstrap procedure as in Chwialkowski et al. [2014]. The reasons for employing the Monte Carlo testing strategy instead of the wild-bootstrap are 1). The non-parametric Stein operator depends on the random function $\widehat{s}_t$ so that classical results for V-statistics convergence which assume that the sole source of randomness is the bootstrap may not apply[2]. 2). While the wild-bootstrap is asymptotically consistent as observed sample size $n \to \infty$, it may not necessarily control the type-I error in a non-asymptotic regime where n is fixed. More details can be found in Appendix F.1.

Here we note that any test which is based on the summary statistic t will only be able to test for a distribution up to equivalence of their distributions with respect to the summary statistic t; two distributions P and Q are equivalent w.r.t. the summary statistics t if $P(\boldsymbol{X}|t(\boldsymbol{X})) = Q(\boldsymbol{X}|t(\boldsymbol{X}))$. Thus the null hypothesis for the NP-KSD test is that the distribution is equivalent to P with respect to t. Hence, the null hypothesis specifies the conditional distribution, not the unconditional distribution.

Related works To assess whether an implicit generative models can generate samples that are *significantly* consistent with the desired data model, several hypothesis testing procedures have been studied. Jitkrittum et al. [2018] has proposed two kernel-based tests, called Relative Unbiased Mean Embedding (Rel-UME) test and Relative Finite-Set Stein Discrepancy (Rel-FSSD) test, for testing relative model goodness-of-fit, that is, testing whether model S is a better fit to the observations than model R. While Rel-UME is applicable to implicit generative models, Rel-FSSD still requires explicit knowledge of the unnormalised density.

The idea for assessing sample quality for implicit generative models could be viewed as a very imbalanced two-sample problem, where samples generated from the implicit model are compared

[2] A KSD with random Stein kernel has been briefly discussed in Fernández et al. [2020] when the h_q function requires estimation from relevant survival functions.

Algorithm 2 Assessment procedures for implicit generative models

Input: Observed sample set $\mathbb{S} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$; the generator G and generated sample size N; estimation statistics t; RKHS kernel K; re-sampling size B; bootstrap sample size b; confidence level α;

1: Estimate $\widehat{s}(z^{(i)}|t(z^{(-i)}))$ based on Algorithm 1.
2: Uniformly generate re-sampling index $\{i_1, \ldots, i_B\}$ from $[m]$, with replacement.
3: Compute $\tau = \widehat{\text{NP-KSD}}^2(\widehat{s}_t; \mathbb{S})$ in Eq. (14).
4: Simulate $\mathbb{S}_i = \{\boldsymbol{y}'_1, \ldots, \boldsymbol{y}'_n\}$ for $i \in [b]$ from G.
5: Compute $\tau_i = \widehat{\text{NP-KSD}}^2(\widehat{s}_t; \mathbb{S}_i)$ in again with index re-sampling.
6: Estimate the empirical (1- α) quantile $\gamma_{1-\alpha}$ via $\{\tau_1, \ldots, \tau_b\}$.

Output: Reject the null hypothesis if $\tau > \gamma_{1-\alpha}$; otherwise do not reject.

with the observed data. For general two-sample problems, maximum-mean-discrepancy (MMD) could in principle also be applied for assessing sample qualities for implicit models. With an efficient choice of (deep) kernel, Liu et al. [2020] applied MMD tests to assess the distributional difference for image data, e.g. MNIST [LeCun et al., 1998] v.s. digits image trained via deep convolutional GAN (DCGAN) [Radford et al., 2015]; CIFAR10 [Krizhevsky, 2009] v.s. CIFAR10.1 [Recht et al., 2019]. However, we note that the MMD-based two-sample approach for model assessment suffers from two major limitations. Firstly, MMD tests [Gretton et al., 2012a] require a balanced sample size for two sets of samples; if n_1 and n_2 are the sizes for each sample set, then MMD tests have well-controlled type-I error when $n_1 \approx n_2$. In the model assessment setting, we may observe a small fixed sample set n while being able to generate $N \gg n$ samples from the generator. The second limitation concerns the idea to use matched sample size in order to enjoy the theoretical guarantees which are available for MMD tests, in which case we may only generate $O(n)$ samples from the generator to balance the observed sample size. As such, the empirical estimator for MMD becomes less accurate, which would significantly reduce the test power for small n. This approach would not exploit the fact that the data generator can generate a very large amount of samples if desired. Detailed discussions on MMD two-sample testing procedures can be found in Appendix F.1, including Table 5 showing uncontrolled type-I errors for imbalanced sample size.

4 Experiments

4.1 Baseline and competing approaches

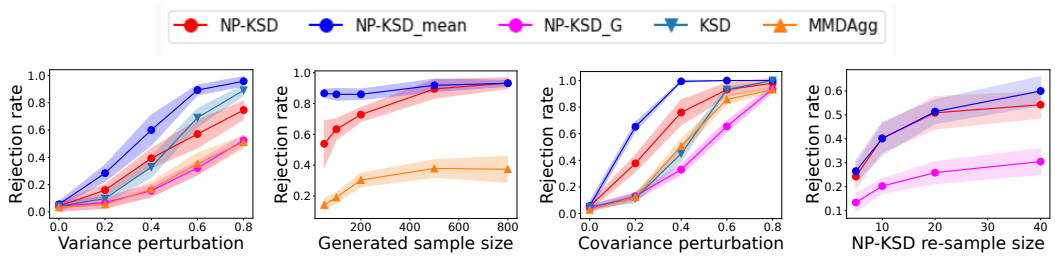
We illustrate the proposed NP-KSD testing procedure with different choice of summary statistics. We denote by **NP-KSD** the version which uses the estimation of the conditional score, i.e. $t(x^{(-i)}) = x^{(-i)}$; by **NP-KSD_mean** the version which uses conditioning on the mean statistics, i.e. $t(x^{(-i)}) = \frac{1}{m-1}\sum_{j \neq i} x^{(j)}$; and by **NP-KSD_G** the version which fits a Gaussian model as conditional density[3].

Two-sample testing methods can be useful for model assessment, where the observed sample set is tested against sample set generated from the model. In our setting where $n \ll N$, we consider a consistent non-asymptotic MMD-based test, **MMDAgg** [Schrab et al., 2021], as our competing approach; see Appendix F.1 for details on computing MMDAgg statistics and discussions on its comparison with the MMD tests [Gretton et al., 2012a][4].

For synthetic distributions where the null models have explicit densities, we include the **KSD** goodness-of-fit testing procedure in Section 2 as the baseline. Gaussian kernels are used and the median heuristic [Gretton et al., 2007] is applied for bandwidth selection. As a caveat, in view of [Gorham and Mackey, 2015], when the kernel decays more rapidly than the score function grows,

[3] **NP-KSD_G** for non-Gaussian densities is generally mis-specified. We deliberately check this case to assess the robustness of the NP-KSD procedure under model mis-specification.

[4] We remark on another desirable property for MMDAgg. MMDAgg is able to (implicitly) select the optimal kernel parameter to achieve better test power without *data-splitting*, where data-splitting results into smaller sample size available for performing the test, thus reducing data efficiency. However, this property is not the reason we choose MMDAgg as our baseline here; we choose it because MMDAgg is a non-asymptotic MMD-based test which is capable of handling the an imbalanced sample situation, i.e. $n \ll N$.



(a) GVD: $n = 100$ (b) GVD:$\sigma_{per} = 0.4$ (c) MoG: $n = 200$ (d) MoG: $m = 40$

Figure 1: Rejection rates of the synthetic distributions: test level $\alpha = 0.05$; 100 trials per round of experiment; 10 rounds of experiment are taken for average and standard deviation; bootstrap sample size $b = 500$; $m = 3$ for (a) and (b); $m = 6$ for (c); $n = 100$, $\sigma_{per} = 0.5$ for (d).

then identifiability of q_t through a KSD method may not be guaranteed. We note that MMD tests do not have controlled type-I error when $n \ll N$, thus are not suitable in this setting. While MMD-based methods to compare and criticise two generative models have been explored [Lloyd and Ghahramani, 2015, Sutherland et al., 2017], MMD tests [Gretton et al., 2012a] are therefore not included in the comparison list. Detailed discussions and illustrations are found in Appendix F.1.

4.2 Experiments on synthetic distributions

Gaussian Variance Difference (GVD) We first consider a standard synthetic setting, studied in Jitkrittum et al. [2017], in which the null distribution is multivariate Gaussian with mean zero and identity covariance matrix. The alternative is set to perturb the the diagonal terms of the covariance matrix, i.e. the variances, all by the same amount.

The rejection rate against the variances perturbation is shown in Figure 1(a). From the result, we see that all the tests presented have controlled type-I error. For all the tests the power increases with increased perturbation. **NP-KSD** and **NP-KSD_mean** outperform the **MMDAgg** approach. Using the mean statistics, **NP-KSD_mean** is having slightly higher power than **KSD**. The mis-specified **NP-KSD_G** has lower power, but is still competitive to **MMDAgg**.

The test power against the sample size N generated from the null model is shown in Figure 1(b). The generated samples are used as another sample set for the **MMDAgg** two-sample procedure, while used for estimating the conditional score for NP-KSD-based methods. As the generated sample size increases, the power of **MMDAgg** increases more slowly than that of the NP-KSD-based methods, which achieve maximum test power in the presented setting. The NP-KSD-based tests tend to have lower variability of the test power, indicating more reliable testing procedures than **MMDAgg**.

Mixture of Gaussian (MoG) Next, we consider as a more difficult problem that the null model is a two-component mixture of two independent Gaussians. Both Gaussian components have identity covariance matrix. The alternative is set to perturb the covariance between adjacent coordinates.

The rejection rates against this perturbation of covariance terms are presented in Figure 1(c). The results show controlled type-I error. The **NP-KSD** and **NP-KSD_mean** tests have better test power compared to **KSD** and **MMDAgg**, although **NP-KSD** has slightly higher variance. Among the NP-KSD tests, the smallest variability is achieved by **NP-KSD_mean**. For the test with $m = 40$, we also vary the re-sample size B. As shown in Figure 1(d), the variability of the average test power increased slightly. From the result, we also see that for $B = 20 = m/2$ the test power is already competive compared to $B = 40$. Additional experimental results including computational runtime and training generative models for synthetic distributions are included in Appendix C.

4.3 Applications to deep generative models

For real-world applications, we assess models trained from well-studied generative modelling procedures, including a Generative Adversarial Network (**GAN**) [Goodfellow et al., 2014] with multilayer perceptron (MLP), a Deep Convolutional Generative Adversarial Network (**DCGAN**) [Radford et al., 2015], and a Variational Autoencoder (**VAE**) [Kingma and Welling, 2013]. We also consider a Noise

	GAN_MLP	DCGAN	VAE	NCSN	Real
NP-KSD	1.00	0.92	1.00	1.00	0.03
NP-KSD_m	1.00	1.00	1.00	1.00	0.01
MMDAgg	1.00	0.73	0.93	1.00	0.06

Table 1: Rejection rate for MNIST generative models.

	DCGAN	NCSN	CIFAR10.1	Real
NP-KSD	0.68	0.73	0.92	0.06
NP-KSD_m	0.74	0.81	0.96	0.02
MMDAgg	0.48	0.57	0.83	0.07

Table 2: Rejection rate for CIFAR10 generative models.

Conditional Score Network (**NCSN**) [Song and Ermon, 2020], which is a score-based generative modelling approach, where the score functions are learned [Song and Ermon, 2019] to performed annealed Langevin dynamics for sample generation. We also denote **Real** as the scheme that generates samples randomly from the training data, by using a held-out set of samples from the real data, and which hence can be viewed as an implicit generator of the null distribution.

MNIST Dataset This dataset contains 28×28 grey-scale images of handwritten digits [LeCun et al., 1998][5]. It consist of $60,000$ training samples and $10,000$ test samples. Deep generative models in Table 1 are trained using the training samples. We assess the quality of these trained generative models by testing against the true observed MNIST samples (from the test set). Samples from both distributions are visually illustrated in Figure 3 in Appendix C. For our experiments, 600 samples are generated from the generative models and 100 samples are used for the test;the samples are then down-sampled into 7×7 images, as in Schrab et al. [2021]. The test level is $\alpha = 0.05$. From Table 1, we see that all the deep generative models have high rejection rates. Testing with the **Real** scheme has controlled type-I error. Thus, NP-KSD detects that the "real" data are a true sample set from the underlying dataset, but often rejects the synthetic data generators.

CIFAR10 Dataset This dataset contains 32×32 RGB coloured images [Krizhevsky, 2009][6]. It consist of $50,000$ training samples and $10,000$ test samples. Deep generative models in Table 2 are trained using the training samples, and test samples are randomly drawn from the test set. Samples are illustrated in Figure 4 in Appendix C. We also compare with the CIFAR10.1 dataset [Recht et al., 2018][7], which is created to differ from CIFAR10 to investigate generalisation power of training classifiers. For our experiments, 800 samples are generated from the generative models and 200 samples are used for the test; the samples are then down-sampled into 8×8 images, in a similar fashion as in Schrab et al. [2021]. The test level is $\alpha = 0.05$. Table 2 shows higher rejection rates for NP-KSD tests compared to MMDAgg, echoing the results for synthetic distributions. The trained **DCGAN** generates samples with lower rejection rate in the CIFAR10 dataset than in the CIFAR10.1 dataset. We also see that the score-based NCSN has higher rejection rate than the non-score-based DCGAN, despite NP-KSD being a score-based test. The distribution difference between CIFAR10 and CIFAR10.1 can be well-distinguished from the tests. Testing with the **Real** scheme again has controlled type-I error.

5 Conclusion and future directions

Synthetic data are in high demand, for example for training ML procedures; quality is important. Synthetic data which miss important features in the data can lead to erroneous conclusions, which for example in medical applications could be fatal, and in loan applications could be detrimental to personal or business development. NP-KSD provides a method for assessing synthetic data generators

[5] `https://pytorch.org/vision/main/generated/torchvision.datasets.MNIST.html`

[6] `https://pytorch.org/vision/stable/generated/torchvision.datasets.CIFAR10.html`

[7] `https://github.com/modestyachts/CIFAR-10.1/tree/master/datasets`

which comes with theoretical guarantees. Our experiments on synthetic data have shown that NP-KSD achieves good test power and controlled type-I error. On real data, NP-KSD detects samples from the true dataset. That none of the classical deep learning methods used in this paper has a satisfactory rejection rate indicates scope for further developments in synthetic data generation.

Future research will assess alternatives to the computer-intensive Monte Carlo method for estimating the null distribution, for example through adapting wild-bootstrap procedures. It will explore alternative choices of score estimation as well as of kernel functions.

Finally, some caution is advised. The choice of summary statistic may have strong influence on the results and a classification based on NP-KSD may still miss some features. Erroneous decisions could be reached when training classifiers. Without scrutiny this could lead to severe consequences for example in health science applications. Future work will devote more attention on analysing the choice of summary statistic. Yet NP-KSD is an important step towards understanding black-box data generating methods and thus understanding their potential shortcomings.

Acknowledgements

The authors thank the anonymous reviewers for their constructive comments and suggestions to improve the paper. G.R. and W.X. acknowledge the support from EPSRC grant EP/T018445/1. G.R is also supported in part by EPSRC grants EP/W037211/1, EP/V056883/1, and EP/R018472/1.

References

AD Barbour and LHY Chen. An Introduction to Stein's method. *Lecture Notes Series. Institute for Mathematical Sciences. National University of Singapore*, 4, 2005.

Alessandro Barp, Francois-Xavier Briol, Andrew Duncan, Mark Girolami, and Lester Mackey. Minimum Stein discrepancy estimators. In *Advances in Neural Information Processing Systems*, pages 12964–12976, 2019.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

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

Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit. In *International Conference on Machine Learning*, pages 2606–2615. PMLR, 2016.

Kacper P Chwialkowski, Dino Sejdinovic, and Arthur Gretton. A wild bootstrap for degenerate kernel tests. In *Advances in Neural Information Processing Systems*, pages 3608–3616, 2014.

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1): 53–65, 2018.

Tamara Fernández, Wenkai Xu, Marc Ditzhaus, and Arthur Gretton. A kernel test for quasi-independence. *Advances in Neural Information Processing Systems*, 33:15326–15337, 2020.

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*, 27, 2014.

Jackson Gorham and Lester Mackey. Measuring sample quality with Stein's method. In *Advances in Neural Information Processing Systems*, pages 226–234, 2015.

Jackson Gorham and Lester Mackey. Measuring sample quality with kernels. In *International Conference on Machine Learning*, pages 1292–1301. PMLR, 2017.

Jackson Gorham, Anant Raj, and Lester Mackey. Stochastic Stein discrepancies. *Advances in Neural Information Processing Systems*, 33:17931–17942, 2020.

A. Gretton, K. Fukumizu, C.H. Teo, L. Song, B. Schölkopf, and A.J. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems*, pages 585–592, 2008.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, pages 513–520, 2007.

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

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 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. In *Advances in Neural Information Processing Systems*, pages 1205–1213, 2012b.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

Susan Holmes and Gesine Reinert. Stein's method for the bootstrap. In *Stein's Method*, volume 46, pages 93–133. Institute of Mathematical Statistics, 2004.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709, 2005.

Wittawat Jitkrittum, Zoltán Szabó, Kacper P Chwialkowski, and Arthur Gretton. Interpretable distribution features with maximum testing power. In *Advances in Neural Information Processing Systems*, pages 181–189, 2016.

Wittawat Jitkrittum, Wenkai Xu, Zoltán Szabó, Kenji Fukumizu, and Arthur Gretton. A linear-time kernel goodness-of-fit test. In *Advances in Neural Information Processing Systems*, pages 262–271, 2017.

Wittawat Jitkrittum, Heishiro Kanagawa, Patsorn Sangkloy, James Hays, Bernhard Schölkopf, and Arthur Gretton. Informative features for model comparison. In *Advances in Neural Information Processing Systems*, pages 808–819, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

Yann LeCun, Lawrence D Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Müller, Eduard Säckinger, Patrice Simard, and Vladimir Vapnik. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261(276):2, 1995.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting Structured Data*, 1(0), 2006.

A. J. Lee. *U-Statistics: Theory and Practice*. CRC Press, 1990.

Christophe Ley, Gesine Reinert, and Yvik Swan. Stein's method for comparison of univariate distributions. *Probability Surveys*, 14:1–52, 2017.

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*, 30, 2017.

Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J Sutherland. Learning deep kernels for non-parametric two-sample tests. In *International Conference on Machine Learning*, pages 6316–6326. PMLR, 2020.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized Stein discrepancy for goodness-of-fit tests. In *International Conference on Machine Learning*, pages 276–284, 2016.

James Robert Lloyd and Zoubin Ghahramani. Statistical model criticism using kernel two sample tests. In *NeurIPS*, pages 829–837, 2015.

Guillaume Mijoule, Gesine Reinert, and Yvik Swan. Stein's density method for multivariate continuous distributions. *arXiv preprint arXiv:2101.05079*, 2021.

Frédéric Ouimet. General formulas for the central and non-central moments of the multinomial distribution. *Stats*, 4(1):18–27, 2021.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

Gesine Reinert. Three general approaches to Stein's method. *An introduction to Stein's method*, 4: 183–221, 2005.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in Neural Information Processing Systems*, 29, 2016.

Antonin Schrab, Ilmun Kim, Mélisande Albert, Béatrice Laurent, Benjamin Guedj, and Arthur Gretton. MMD aggregated two-sample test. *arXiv preprint arXiv:2110.15073*, 2021.

Antonin Schrab, Benjamin Guedj, and Arthur Gretton. KSD aggregated goodness-of-fit test. *arXiv preprint arXiv:2202.00824*, 2022.

David W Scott and Stephan R Sain. Multidimensional density estimation. *Handbook of statistics*, 24: 229–261, 2005.

Raghav Singhal, Xintian Han, Saad Lahlou, and Rajesh Ranganath. Kernelized complete conditional stein discrepancy. *arXiv preprint arXiv:1904.04478*, 2019.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in Neural Information Processing Systems*, 33:12438–12448, 2020.

Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR, 2020.

Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12(Jul): 2389–2410, 2011.

Charles Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*. The Regents of the University of California, 1972.

Danica J Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alexander J Smola, and Arthur Gretton. Generative models and model criticism via optimized maximum mean discrepancy. In *ICLR (Poster)*, 2017.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Ilya Tolstikhin, Bharath K Sriperumbudur, and Krikamol Muandet. Minimax estimation of kernel mean embeddings. *The Journal of Machine Learning Research*, 18(1):3002–3048, 2017.

Aad W Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge University Press, 2000.

Dilin Wang, Zhe Zeng, and Qiang Liu. Stein variational message passing for continuous graphical models. In *International Conference on Machine Learning*, pages 5219–5227. PMLR, 2018.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

Li Wenliang, Danica J Sutherland, Heiko Strathmann, and Arthur Gretton. Learning deep kernels for exponential family densities. In *International Conference on Machine Learning*, pages 6737–6746. PMLR, 2019.

Wenkai Xu. Standardisation-function kernel Stein discrepancy (Sf-KSD): A unifying approach for non-parametric goodness-of-fit testing. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Wenkai Xu and Gesine Reinert. A Stein goodness-of-test for exponential random graph models. In *International Conference on Artificial Intelligence and Statistics*, pages 415–423. PMLR, 2021.

Wenkai Xu and Gesine Reinert. AgraSSt: Approximate graph Stein statistics for interpretable assessment of implicit graph generators. *arXiv preprint arXiv:2203.03673*, 2022.

Yuhao Zhou, Jiaxin Shi, and Jun Zhu. Nonparametric score estimators. In *International Conference on Machine Learning*, pages 11513–11522. PMLR, 2020.

Jingwei Zhuo, Chang Liu, Jiaxin Shi, Jun Zhu, Ning Chen, and Bo Zhang. Message passing Stein variational gradient descent. In *International Conference on Machine Learning*, pages 6018–6027. PMLR, 2018.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] abstract and Section 1
   (b) Did you describe the limitations of your work? [Yes] Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Section 5
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Section 3
   (b) Did you include complete proofs of all theoretical results? [Yes] In supplementary material
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code attached in supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Section 4. Additional details are included in Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Section 4
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Appendix C
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Section 4
   (b) Did you mention the license of the assets? [N/A] Open source datasets are cited
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary Information for A Kernelised Stein Statistic for Assessing Implicit Generative Models

A Justification of the Stein operator

Here we justify the two-step approach for constructing a Stein operator.

A.1 Step 1: A non-parametric Stein operator

Suppose we can estimate the conditional distribution from data. Then we can create a Markov chain with values in $(\mathbb{R}^d)^m$ as follows. Starting with $Z_0 = \{x_1, \ldots, x_m\}$ with $x_i \in \mathbb{R}^d$ for $i = 1, \ldots, m$ (often we choose $d = 1$), we pick an index $I \in [m]$ at random. If $I = i$ we replace x_i by X_i' drawn from the conditional distribution of X_i given $(X_j : j \neq i)$. This gives $Z_1 = (x_1, \ldots, x_{i-1}, X_i', x_{i+1}, \ldots, x_m)$[8] see for example Reinert [2005]. To make this a continuous-time Markov process generator, we wait an exponential(1)-distributed time before every change.

This generator induces a Stein operator for the target distribution as follows. Here we take $d = 1$ for clarity; the generalisation to other d is straightforward. Let $f : \mathbb{R}^m \to \mathbb{R}$ and consider the expectation w.r.t. the one-step evolution of the Markov chain

$$\begin{aligned}&\mathbb{E}_{-i}[f(x^{(1)}, \ldots, x^{(i-1)}, X^{(i)}, x^{(i+1)}, \ldots, x^{(m)})] \\ &= \int f(x^{(1)}, \ldots, x^{(i-1)}, y, x^{(i+1)}, \ldots, x^{(m)}) \mathbb{P}(X^{(i)} = y | X^{(j)} = x^{(j)}, j \neq i).\end{aligned}$$

We now consider the index i as the i-th coordinate of multivariate random variables in $\mathbb{R}^m$. The conditional expectation here, fixing all but the i-th *coordinate* term, only depends on the univariate conditional distribution $Q^{(i)}$ given by $Q^{(i)}(A) = \mathbb{P}(X^{(i)} \in A | X^{(j)} = x^{(j)}, j \neq i)$. Thus, the Stein operator induced from the Markov chain has the form

$$\mathcal{A} f(z) = \mathcal{A}^{(I)} f(z) \tag{16}$$

where

$$\mathcal{A}^{(i)} f(\boldsymbol{x}) = \mathbb{E}_{-i}[f(x^{(1)}, \ldots, x^{(i-1)}, X^{(i)}, x^{(i+1)}, \ldots, x^{(m)})] - f(\boldsymbol{x}). \tag{17}$$

From the law of total expectation it follows that the Stein identity holds.

A.2 Step 2: marginal Stein operators

In Eq. (17), the expectation

$$\mathbb{E}_{-i}[f(x^{(i)}, \ldots, x^{(i-1)}, X^{(i)}, x^{(i+1)}, \ldots, x^{(m)})] - f(x^{(1)}, \ldots, x^{(m)})$$

can itself be treated via Stein's method, by finding a Stein operator $\mathcal{T}^{(i)}$ and a function g such that $g = g_f$ solves the $\mathcal{T}^{(i)}$-Stein equation Eq. (1) for f;

$$\mathcal{T}^{(i)} g(x) = \mathbb{E}_{-i}[f(x^{(1)}, \ldots, x^{(i-1)}, X^{(i)}, x^{(i+1)}, \ldots, x^{(m)})] - f(x^{(1)}, \ldots, x^{(m)}). \tag{18}$$

Fixing $x_j, j \neq i$ and setting $f^{(i)}(x) = f(x^{(1)}, \ldots, x^{(i-1)}, x, x^{(i+1)}, \ldots, x^{(m)})$, we view $\mathcal{T}^{(i)}$ as a Stein operator for a univariate distribution, acting on functions $g = g_{x^{(-i)}} : \mathbb{R} \to \mathbb{R}$.

Summarising the approach, the Stein operator $\mathcal{A}$ acting on functions $f : \mathbb{R}^m \to \mathbb{R}$ underlying the non-parametric Stein operator is

$$\mathcal{A} f(x^{(1)}, \ldots, x^{(m)}) = \mathcal{T}^{(I)} g_{f, x^{-I}}(x^{(I)}) \tag{19}$$

where $I \in [m]$ is a randomly chosen index. In view of Eq. (19) we take $g : \mathbb{R}^m \to \mathbb{R}$, write $g_{x^{(-i)}}(x) : \mathbb{R} \to \mathbb{R}$ for the univariate function which acts only on the coordinate i and fixes the other

[8] Denote $Z_1 = (x^{(1)}, \ldots, x^{(i-1)}, X^{(i)\prime}, x^{(i+1)}, \ldots, x^{(m)}) \in \mathbb{R}^m$ where the superscript (i) is used for the coordinate index.

coordinates to equal $x^{(-i)}$, we as Stein operator (using the same letter $\mathcal{A}$ as before, which is abuse of notation);

$$\mathcal{A}g(x^{(1)}, \ldots, x^{(m)}) = \mathcal{T}^{(I)} g_{x^{-I}}(x^{(I)}).$$

This formulation simplifies Eq. (19) in that we no longer have to consider the connection between f and g. The final step is to note that when we condition on the random index I, again a Stein operator is obtained, as follows. As

$$\mathbb{E}_I[\mathcal{A}g(x^{(1)}, \ldots, x^{(m)})] = \frac{1}{m} \sum_{i=1}^{m} \mathcal{T}^{(i)} g_{x^{(-i)}}(x^{(i)}). \tag{20}$$

As $\mathbb{E}[\mathcal{T}^{(i)} g_{X^{(-i)}}(X^{(i)})] = 0$, the Stein identity is satisfied. The operator in Eq. (20) is the Stein operator given in Eq. (7).

The strategy of averaging over all coordinate terms $i \in [m]$ has also studied in variational inference, via coordinate ascent variational inference (CAVI); see for example Bishop and Nasrabadi [2006] which focuses on latent variable inference.

B Proofs and additional results

Assuming that if $f \in \mathcal{H}$ then $-f \in \mathcal{H}$ we can assume that the supremum over the expectation is non-negative, and with Eq. 12,

$$\begin{aligned} 0 \leq \text{NP-KSD}_t(P\|Q) &= \sup_{f \in \mathcal{B}_1(\mathcal{H})} \mathbb{E}_p[\widehat{\mathcal{A}}^B_{t,N} f] \\ &= \sup_{f \in \mathcal{B}_1(\mathcal{H})} \{\mathbb{E}_p \mathcal{A}_t f + \mathbb{E}_p[\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t] f\} \\ &= \sup_{f \in \mathcal{B}_1(\mathcal{H})} \{\mathbb{E}_p \mathcal{A}_t f + \mathbb{E}_p[\widehat{\mathcal{A}}^B_{t,N} - \widehat{\mathcal{A}}_{t,N}] f + \mathbb{E}_p[f(\widehat{s}^{(i)}_{t,N} - \mathbf{s}_t)]\}. \end{aligned} \tag{21}$$

Here $\widehat{\mathcal{A}}_{t,N}$ is the Stein operator using the estimated conditional score function $\hat{s}_{t,N}$ with the estimation based on N synthetic observations. We now assess the contribution to Eq. 21 which stems from estimating the score function. Note that here we only need to estimate a one-dimensional score function and hence the pitfalls of score estimation in high dimensions do not apply. We note however the contribution of Zhou et al. [2020] for a general framework.

Assume that we estimate the univariate conditional density $q_t^{(i)}$ based on N samples. We assume that $q_t^{(i)}$ is differentiable, and we denote its score function by

$$s_t^{(i)}(x) = \frac{(q_t^{(i)})'(x)}{q_t^{(i)}(x)}.$$

We next prove an extension of Proposition 3.1.

Proposition B.1. *Suppose that for $i \in [m]$, $\widehat{s}_N^{(i)}$ is a consistent estimator of the univariate score function $s^{(i)}$. Let $\mathcal{T}^{(i)}$ be a Stein operator for the univariate differentiable probability distribution $Q^{(i)}$ of the generalised density operator form Eq. (8). Let*

$$\begin{aligned} \widehat{\mathcal{T}}_N^{(i)} g(x) &= g'(x) + g(x)\widehat{s}_N^{(i)} \\ \widehat{\mathcal{A}} g(\boldsymbol{x}) &= \widehat{\mathcal{T}}_N^{(I)} g_{x^{(-I)}}(x^{(I)}) \qquad \textit{and} \\ \widehat{\mathcal{A}}_N g(\boldsymbol{x}) &= \frac{1}{m} \sum_{i \in [m]} \widehat{\mathcal{T}}_N^{(i)} g_{x^{(-i)}}(x^{(i)}). \end{aligned}$$

Then $\widehat{\mathcal{T}}_N^{(i)}$ is a consistent estimator for $\mathcal{T}^{(i)}$, and $\widehat{\mathcal{A}}$ as well as $\widehat{\mathcal{A}}_N$ are consistent estimators of $\mathcal{A}$.

Proof. Take a fixed x. As $\widehat{s}_N^{(i)}$ is a consistent estimator of $s^{(i)}$, it holds that for any $\epsilon > 0$ and for any x in the range of $s^{(i)}$,

$$\mathbb{P}(|\widehat{s}_N^{(i)}(x) - s^{(i)}(x)| > \epsilon) \to 0$$

as $N \to \infty$. Here ω denotes the random element for the estimation, which is implicit in $\widehat{q}_N$. On the set

$$A_\epsilon = \left\{ |\widehat{s}_N^{(i)}(x) - s^{(i)}(x)| \leq \epsilon \right\}$$

we have that

$$|\widehat{\mathcal{T}}_N^{(i)} g(x) - \mathcal{T}^{(i)} g(x)| \leq \epsilon f(x).$$

For every fixed x this expression tends to 0 as $\epsilon \to 0$. Hence consistency of $\widehat{T}_N$ follows. The last two assertions follow immediately from Eq. 19 and Eq. 20. □

B.1 Asymptotic behaviour of NP-KSD

Here we assess the asymptotic behaviour of NP-KSD^2. With $\boldsymbol{s}_t$ denoting the conditional score function,

$$\text{NP-KSD}_t^2(G\|p) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim p} \left\langle \mathcal{A}_{Q_t} k(\boldsymbol{x},\cdot), \mathcal{A}_{Q_t} k(\boldsymbol{x}',\cdot) \right\rangle_{\mathcal{H}}$$

where $\mathcal{A}_{Q_t} k(\boldsymbol{x},\cdot) = \mathcal{A}_t k(\boldsymbol{x},\cdot)$ can be written as

$$\begin{aligned} \mathcal{A}_t k(\boldsymbol{x},\cdot) &= \frac{1}{m} \sum_{i\in[m]} \left\{ \mathcal{A}_{\widehat{Q}_t^{(i)}} k(\boldsymbol{x},\cdot) + k(\boldsymbol{x},\cdot)(\widehat{s}_{t,N}^{(i)} - s_t^{(i)}) \right\} \\ &= \frac{1}{m} \sum_{i\in[m]} \left\{ \frac{\partial}{\partial x^{(i)}} k(\boldsymbol{x},\cdot) + k(\boldsymbol{x},\cdot)\widehat{s}_{t,N}^{(i)} + k(\boldsymbol{x},\cdot)(\widehat{s}_{t,N}^{(i)} - s_t^{(i)}) \right\}. \end{aligned} \tag{22}$$

Recall that KSD^2 is given in Eq. (5) by

$$\text{KSD}^2(q\|p) = \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle \mathcal{A}_q k(\boldsymbol{x},\cdot), \mathcal{A}_q k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}}],$$

where $\text{KSD}(q\|p)$ is a deterministic quantity which under weak assumption vanishes when $p = q$. Moreover,

$$\text{KSD}_t^2(q_t\|p) = \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle \mathcal{A}_t k(\boldsymbol{x},\cdot), \widehat{\mathcal{A}}_t k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}}.$$

Disentangling this expression in general is carried out using Eq. (7).

Remark B.2. For a Gaussian kernel $k = k_G$ as used in this paper, we can exploit its factorisation;

$$k_G(\boldsymbol{x},\tilde{\boldsymbol{x}}) = \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^m \left(x^{(i)} - \tilde{x}^{(i)} \right)^2 \right\} = \prod_{i=1}^m \exp\left\{ -\frac{1}{2\sigma^2} \left(x^{(i)} - \tilde{x}^{(i)} \right)^2 \right\}.$$

In this situation, taking $g_{\boldsymbol{x}}(\cdot) = k_G(\boldsymbol{x},\cdot)$, with $\cdot$ denoting an element in $\mathbb{R}^m$, gives

$$g_{\boldsymbol{x};x^{(-i)}}(\cdot) = \exp\left\{ -\frac{1}{2\sigma^2} \sum_{j:j\neq i}^m \left(x^{(j)} - (\cdot)^{(j)} \right)^2 \right\} \exp\left\{ -\frac{1}{2\sigma^2} \left(x^{(i)} - (\cdot)^{(i)} \right)^2 \right\}.$$

For the operator $\mathcal{T}_q^{(i)}$ in Eq. (8) we have

$$\mathcal{T}_q^{(i)} g_{x^{(-i)}}(\cdot) = \exp\left\{ -\frac{1}{2\sigma^2} \sum_{j=1}^m \left(x^{(j)} - (\cdot)^{(j)} \right)^2 \right\} \left(\frac{1}{\sigma^2}(x^{(i)} - (\cdot)^{(i)}) + (\log q_{t(x^{(-i)})})'(x^{(i)}) \right).$$

Thus, the operator $\mathcal{A}_t$ decomposes as

$$\begin{aligned} \frac{1}{m} \sum_{i=1}^m \mathcal{T}_q^{(i)} g_{x^{(-i)}}(\cdot) &= \exp\left\{ -\frac{1}{2\sigma^2} \sum_{j=1}^m \left(x^{(j)} - (\cdot)^{(j)} \right)^2 \right\} \\ &\quad \frac{1}{m} \sum_{i=1}^m \left\{ \frac{1}{\sigma^2} \left(x^{(i)} - (\cdot)^{(i)} \right) + (\log q_{t(x^{(-i)})})'(x^{(i)}) \right\}. \end{aligned}$$

For a proof of consistency, in our setting the Stein operators are only applied to the observations $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ and hence, for our applications, pointwise consistent estimation suffices, in the sense that that for $i = 1, \ldots, m$, $\widehat{s}^{(i)}_{t,N} = \widehat{s}^{(i)}_{t,N}(x^{(i)})$ is a consistent estimator of the univariate score function $s^{(i)}_t = \{\log(q(x^{(i)})|t(x^{(-i)})\}'$. Score matching estimators often satisfy not only consistency but also asymptotic normality, see for example Song et al. [2020]. Such an assumption is required for Theorem 3.2; recall that we use the notation $\hat{\boldsymbol{s}}_{t,N} = (\hat{s}_{t,N}(x^{(i)}), i \in [m])$. To prove Theorem 3.2 we re-state it for convenience.

Theorem B.3. *Assume that the score function estimator vector $\hat{s}_{t,N}$ is asymptotically normal with mean 0 and covariance matrix $N^{-1}\Sigma_s$. Then* NP-KSD$^2_t(G\|p)$ *converges in probability to* KSD$^2_t(q_t\|p)$ *at rate at least* $\min(B^{-\frac{1}{2}}, N^{-\frac{1}{2}})$.

Proof. We have from Eq. (5)

$$\text{NP-KSD}^2_t(G\|p) = \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle \widehat{\mathcal{A}}^B_{t,N} k(\boldsymbol{x},\cdot), \widehat{\mathcal{A}}^B_{t,N} k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}}.$$

Expanding this expression, with $\mathcal{A}_t$ denoting the score Stein operator in Eq. 7 for the conditional distribution q_t,

$$\begin{aligned}
\text{NP-KSD}^2_t(G\|p) &= \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle \mathcal{A}_t k(\boldsymbol{x},\cdot), \mathcal{A}_t k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}} \\
&\quad + \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle (\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t) k(\boldsymbol{x},\cdot), \mathcal{A}_t k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}} \\
&\quad + \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle \mathcal{A}_t k(\boldsymbol{x},\cdot), (\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t) k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}} \\
&\quad + \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle (\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t) k(\boldsymbol{x},\cdot), \widehat{(}\mathcal{A}^B_{t,N} - \mathcal{A}_t) k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}} \\
&= \text{KSD}^2(q_t\|p) + 2\,\mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle (\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t) k(\boldsymbol{x},\cdot), \mathcal{A}_t k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}} \\
&\quad + \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle (\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t) k(\boldsymbol{x},\cdot), (\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t) k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}}
\end{aligned}$$

where we used the symmetry of the inner product in the last step. Now, for any function g for which the expression is defined,

$$(\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t) g(\boldsymbol{x}) = (\widehat{\mathcal{A}}^B_{t,N} - \widehat{\mathcal{A}}_{t,N}) g(\boldsymbol{x}) + (\widehat{\mathcal{A}}_{t,N} - \mathcal{A}_t) g(\boldsymbol{x}) \tag{23}$$

recalling that $\widehat{\mathcal{A}}_{t,N}$ is the Stein operator using the estimated conditional score function $\hat{s}_{t,N}$ with the estimation based on N synthetic observations.

To analyse Eq. 23 we first consider $(\widehat{\mathcal{A}}_{t,N} - \mathcal{A}_t) g(\boldsymbol{x})$;

$$\widehat{\mathcal{A}}_{t,N} g(\boldsymbol{x}) - \mathcal{A}_t g(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} g(x^{(i)})(\widehat{s}^{(i)}_{t,N}(x^{(i)}) - s^{(i)}_t(x^{(i)})). \tag{24}$$

As for $x^{(1)}, \ldots, x^{(m)}$ the vector $\hat{\boldsymbol{s}}_{t,N} = (\hat{s}_{t,N}(x^{(i)}), i \in [m])$ is assumed to be asymptotically normal with mean 0 and covariance matrix $N^{-1}\Sigma_s$, it follows from Eq. 24 that, asymptotically, $\sqrt{N}(\widehat{\mathcal{A}}_{t,N} g(\boldsymbol{x}) - \mathcal{A}_{q_t} g(\boldsymbol{x}))$ has a multivariate normal distribution and, in particular, $(\widehat{\mathcal{A}}_{t,N} g(\boldsymbol{x}) - \mathcal{A}_{q_t} g(\boldsymbol{x}))$ has fluctuations of the order $N^{-\frac{1}{2}}$.

For the term $(\widehat{\mathcal{A}}^B_{t,N} - \widehat{\mathcal{A}}_{t,N}) g(\boldsymbol{x})$ in Eq. 23 we have

$$\begin{aligned}
(\widehat{\mathcal{A}}^B_{t,N} - \widehat{\mathcal{A}}_{t,N}) g(\boldsymbol{x}) &= \frac{1}{B}\sum_{b=1}^{B} \mathcal{T}_{t,N}{}^{(i_b)} g(x) - \mathcal{A}_{t,N} g(\boldsymbol{x}) \\
&= \sum_{i=1}^{m}\left\{\frac{1}{B}\sum_{b=1}^{B} \mathcal{T}_{t,N}{}^{(i_b)} g(\boldsymbol{x}) \mathbb{1}(i_b = i) - \frac{1}{m}\mathcal{T}_{t,N}{}^{(i)} g(\boldsymbol{x})\right\}.
\end{aligned}$$

Let $k_i = \sum_{b=1}^{B} \mathbb{1}(i_b = i)$ the number of times that i is re-sampled. Then $\mathbb{E}(k_i) = \frac{B}{m}$ and we have

$$\begin{aligned}
(\widehat{\mathcal{A}}^B_{t,N} - \widehat{\mathcal{A}}_{t,N}) g(\boldsymbol{x}) &= \sum_{i=1}^{m} \mathcal{T}^{(i)}_{t,N} g(\boldsymbol{x}) \left\{\frac{1}{B} k_i - \frac{1}{m}\right\} \\
&= \frac{1}{B}\sum_{i=1}^{m} \mathcal{T}^{(i)}_{t,N} g(\boldsymbol{x}) \left\{k_i - \mathbb{E}(k_i)\right\}.
\end{aligned}$$

This term is known to be approximately mean zero normal with finite variance $\Sigma(\hat{\boldsymbol{s}}_{t,N}; g)$ (which depends on $\hat{\boldsymbol{s}}_{t,N}$ and g) of order B^{-1}, see for example Holmes and Reinert [2004], where an explicit bound on the distance to normal is provided. This asymptotic normality holds for the operator given the estimated conditional score function. As the bootstrap samples are drawn independently of the score function estimator, without conditioning, the unconditional distribution is a mixture of normal distributions. For an estimator $\hat{\boldsymbol{s}}_{t,N}$ which is asymptotically normally distributed, the variances $\Sigma(\hat{\boldsymbol{s}}_{t,N}; g)$ will converge to $\Sigma(\boldsymbol{s}_t; g)$.

Thus, with Eq. 23,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t)k(\boldsymbol{x},\cdot), \mathcal{A}_t k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}}] \\
&= \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{\mathcal{A}}^B_{t,N} - \widehat{\mathcal{A}}_{t,N})k(\boldsymbol{x},\cdot), \mathcal{A}_t k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}}] + \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{\mathcal{A}}_{t,N} - \mathcal{A}_t)k(\boldsymbol{x},\cdot), \mathcal{A}_t k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}}]
\end{aligned}$$

with the first term is approximately a variance mixture of mean zero normals tending to 0 in probability at rate at least $B^{-\frac{1}{2}}$ as $B \to \infty$, and the second term approximately a mean zero normal variable tending to 0 in probability at rate at least $N^{-\frac{1}{2}}$ as $N \to \infty$.

It remains to consider

$$\mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t)k(\boldsymbol{x},\cdot), (\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t)k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}}.$$

With Eq. 23 we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t)k(\boldsymbol{x},\cdot), (\widehat{\mathcal{A}}^B_{t,N} - \mathcal{A}_t)k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}} \\
&= \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{\mathcal{A}}^B_{t,N} - \widehat{\mathcal{A}}_{t,N})k(\boldsymbol{x},\cdot), (\widehat{\mathcal{A}}^B_{t,N} - \widehat{\mathcal{A}}_{t,N})k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}} && (25) \\
&\quad + \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{\mathcal{A}}^B_{t,N} - \widehat{\mathcal{A}}_{t,N})k(\boldsymbol{x},\cdot), (\widehat{\mathcal{A}}_{t,N} - \mathcal{A}_t)k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}} && (26) \\
&\quad + \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{\mathcal{A}}_{t,N} - \mathcal{A}_t)k(\boldsymbol{x},\cdot), (\widehat{\mathcal{A}}^B_{t,N} - \widehat{\mathcal{A}}_{t,N})k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}} && (27) \\
&\quad + \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{\mathcal{A}}_{t,N} - \mathcal{A}_t)k(\boldsymbol{x},\cdot), (\widehat{\mathcal{A}}_{t,N} - \mathcal{A}_t)k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}}. && (28)
\end{aligned}$$

In Xu and Reinert [2021], Proposition 2, the following result is shown, using the notation as above.

Let

$$Y = \frac{1}{B^2} \sum_{s,t\in[m]} (k_s k_t - \mathbb{E}(k_s k_t)) h_x(s,t).$$

Assume that h_x is bounded and that $Var(Y)$ is non-zero. Then if Z is mean zero normal with variance $Var(Y)$, there is an explicitly computable constant $C > 0$ such that for all three times continuously differentiable functions g with bounded derivatives up to order 3,

$$|\mathbb{E}[g(Y)] - \mathbb{E}[g(Z)] \leq \frac{C}{B}.$$

Moreover, using Equations (17)-(21) from Ouimet [2021], it is easy to see that $Var(Y)$ is of the order B^{-1}. Hence, Term (25) tends to 0 in probability at rate at least B^{-1}. Similarly, using that the bootstrap sampling is independent of the score function estimation, Terms (26) and (27) tend to 0 in probability at rate at least $(NB)^{-\frac{1}{2}}$. For Term (28), from Eq. (24),

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{\mathcal{A}}_{t,N} - \mathcal{A}_t)k(\boldsymbol{x},\cdot), (\widehat{\mathcal{A}}_{t,N} - \mathcal{A}_t)k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}}] \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle(\widehat{s}^{(i)}_{t,N}(x^{(i)}) - s^{(i)}_t(x^{(i)}))k(x^{(i)},\cdot), (\widehat{s}^{(j)}_{t,N}(x^{(j)}) - s^{(j)}_t(x^{(j)}))k(x_j,\cdot)\rangle_{\mathcal{H}}].
\end{aligned}$$

If $\hat{\boldsymbol{s}}_{t,N}$ is approximately normal as hypothesised, then the inner product is approximately a covariance of order N^{-1}, and hence the overall contribution from Term (28) is of order at most N^{-1}. This finishes the proof. □

C Additional experimental details and results

C.1 Additional experiments

Runtime The computational runtime for all tests are shown in Table 3. **MMDAgg** runtime is also shown as a comparison. From the result, we can see that NP-KSD runs generally slower than the

NP-KSD resampling size	B=5	B=10	B=20	B=40	(MMDAgg)
Runtime(s)	4.65	6.56	8.43	10.44	5.02
Rejection Rate	0.24	0.40	0.51	0.55	0.27

Table 3: Computational runtime for various re-sample size B: observed sample size $n = 100$; bootstrap size $b = 200$; dimension $m = 40$. The rejection rate is used for power comparison; higher rejection rates indicate higher power.

permutation-based test **MMDAgg**. This is mainly due to the learning of conditional score functions and the Monte-Carlo based bootstrap procedure. As the re-sample size B increases, NP-KSD test requires longer runtime. However, the rejection rate $B = 20$ is close to that of $B = 40$ (echoing the results shown in Figure 1(d)). While **MMDAgg** generally has faster computational runtime, it has lower test power, which is only comparable to that of NP-KSD with $B = 5$, at which the runtime advantage is not that obvious.

Training on synthetic distributions We also train the deep generative models on the synthetic distributions studied in Section 4 and perform model assessment on the trained models. We consider the standard Gaussian and Mixture of two-component Gaussian problems. We train a generative adversarial network with multi-layer perceptron (**GAN_MLP**)[9] and a variational auto-encoder (VAE) [Kingma and Welling, 2013]. A Noise-Contrastive Score Network **NCSN** is also trained to learn the score function, followed by annealed Langevin dynamics [Song and Ermon, 2019, 2020]. All trainin is carried out via the optimiser Adam [Kingma and Ba, 2014] with adaptive learning rate. The rejection rates are reported in Table 4.

As shown in Table 4, the unimodal Gaussian distribution is easier to be learned by the generative modelling procedures, as compared to the two-component Mixture of Gaussian (MoG) model. As a result, the $\mathrm{NP\text{-}KSD_m}$ testing procedure shows a higher rejection rate on trained MoG generative models compared to that of Gaussian.[10]. However, as these deep models are not designed for training and sampling a simple low-dimensional distribution, it is not surprising that the procedure produce samples that not pass the $\mathrm{NP\text{-}KSD}$ tests.

Inspired by the settings in Gorham and Mackey [2017], where KSD is used to measure sample quality, we also apply $\mathrm{NP\text{-}KSD}$ tests on the Stochastic Gradient Langevin Dynamics (SGLD) [Welling and Teh, 2011] sampling procedure studied in Gorham and Mackey [2017]; in Gorham and Mackey [2017], SGLD is referred to as Stochastic Gradient Fisher Scoring (SGFS). SGLD is capable of sampling unimodal distributions, while it can have problems sampling multimodal data. The rejection rates shown in Table 4 are slightly higher than the test level for MoG, while the type-I error is well controlled for the Gaussian case. Generated samples from SGLD are visualised Figure 2, illustrating that the SGLD samples look plausible for the Gaussian model, but less so for the MoG model.

	GAN_MLP	VAE	NCSN	SGLD	Real
Gaussian	0.36	0.61	0.25	0.06	0.03
MoG	0.78	0.92	0.45	0.12	0.04

Table 4: $\mathrm{NP\text{-}KSD_m}$ rejection rate: observed sample size $n = 100$; bootstrap size is 200. Here a low rejection rate indicates a good type-1 error. Among deep generative models NCSN performs best on both tasks but it still has a very high rejection rate. SGLD outperforms the deep generative models.

[9]DCGAN studied in the main text is particularly useful for the (high-dimensional) image dataset due to the convolutional neural network (CNN) layers; DCGAN is not applicable for the problem in $\mathbb{R}^2$.

[10]We note that $\mathrm{NP\text{-}KSD}$ and $\mathrm{NP\text{-}KSD_m}$ with summary statistics taken to be the mean are equivalent in the two-dimensional problem.

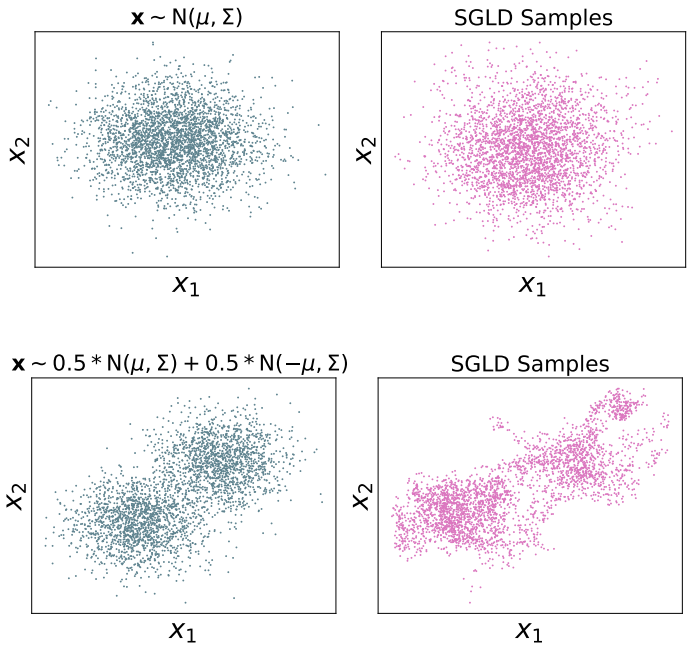
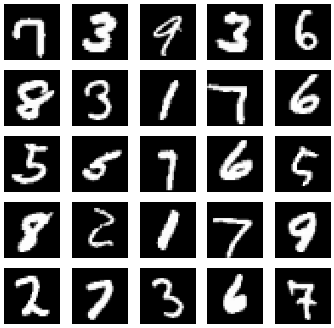


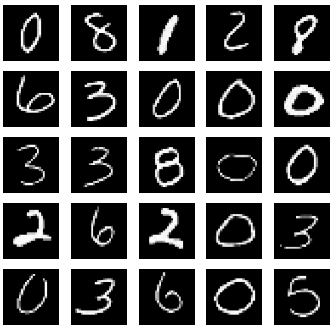
Figure 2: Visualisation of samples generated from Stochastic Gradient Langevin Dynamics (SGLD); top: Gaussian model, bottom: MoG model.

C.2 Data visualisation

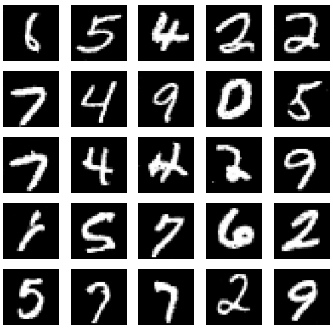
In Figure 3 and Figure 4 we show samples from the MNIST and CIFAR10 dataset, together with samples from trained generative models, respectively.



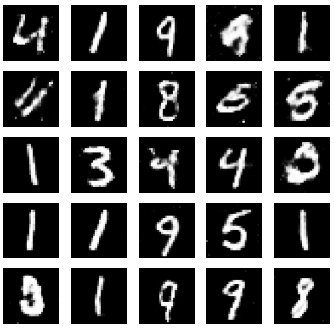
(a) Real MNIST samples



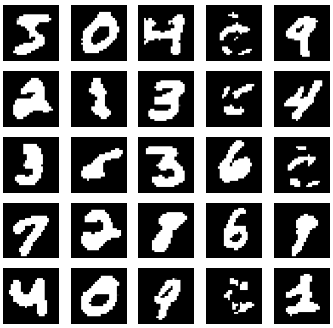
(b) NCSN samples



(c) DCGAN samples

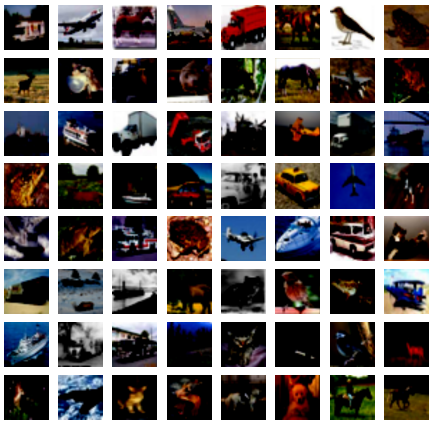


(d) GAN samples

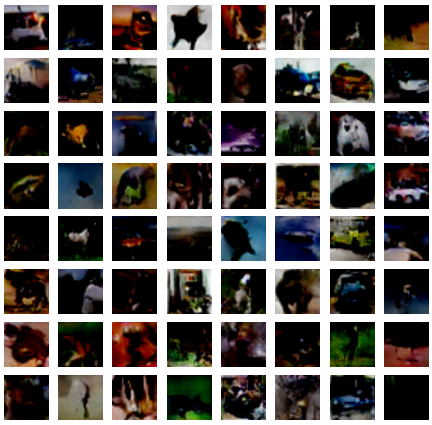


(e) VAE samples

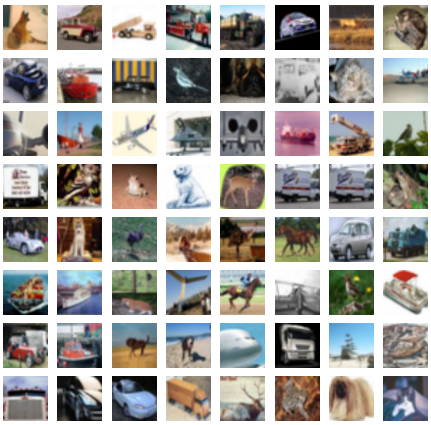
Figure 3: MNIST samples



(a) Real samples



(b) DCGAN samples



(c) CIFAR10.1 samples

(d) NCSN samples

Figure 4: CIFAR10 samples

D Equivalence to the multivariate score-Stein operator

Here we show that the operator in Eq. 7 and the corresponding multivariate score-Stein operator in Eq. 2 are equivalent, when they exist; the difference being the factor $\frac{1}{m}$. First we recall the set-up for score-Stein operators. Let q with smooth support Ω_q be differentiable. The score function of q is the function

$$\mathbf{s}_q = \mathcal{T}_{\nabla,q} 1 = \nabla \log q = \frac{\nabla q}{q}$$

(with the convention that $\mathbf{s}_q \equiv 0$ outside of Ω_q). The score-Stein operator is the vector-valued operator

$$\mathcal{A}_q = \nabla + \mathbf{s}_q \mathrm{I}_m \tag{29}$$

acting on differentiable functions $g : \mathbb{R}^m \to \mathbb{R}$, with I_m denoting the $m \times m$ identity matrix.

Proposition D.1. *When they exist, then the operators in Eq. 7 and in Eq. 2 differ only by a factor* $\frac{1}{m}$.

Proof. Writing ∂_i for the derivative in direction $x^{(i)}$, the score operator acting on differentiable functions $g : \mathbb{R}^m \to \mathbb{R}$ can be written as

$$\mathcal{A}_p g(x) = \sum_{i=1}^{m} \left\{ \partial_i g(x) + g(x) \partial_i (\log q(x)) \right\}. \tag{30}$$

Now, for $i \in [m]$,

$$q(x) = q(x^{(i)} | x^{(j)}, j \neq i) q(x^{(j)}, j \neq i)$$

and hence

$$\partial_i (\log q(x)) = \partial_i \log q(x^{(i)} | x^{(j)}, j \neq i).$$

The assertion follows. □

Example D.2 (Bi-variate Gaussian)**.** Consider $x = (x^{(1)}, x^{(2)})^\top \in \mathbb{R}^2$, i.e. $m = 2$ and $x \sim \mathcal{N}(\mu, \Sigma)$ where $\mu = (\mu^{(1)}, \mu^{(2)})^\top \in \mathbb{R}^2$, $\Sigma = \begin{pmatrix} 1 & \sigma \\ \sigma & 1 \end{pmatrix}$. With the corresponding precision matrix $\Sigma^{-1} = \frac{1}{1-\sigma^2} \begin{pmatrix} 1 & -\sigma \\ -\sigma & 1 \end{pmatrix}$, it is easy to check $Q^{(1)}(X^{(1)} | X^{(2)} = x^{(2)}) \sim \mathcal{N}(\mu^{(1)} + \sigma(x^{(2)} - \mu^{(2)}), 1 - \sigma^2)$. For a bi-variate differentiable test function $g : \mathbb{R}^2 \to \mathbb{R}$, applying the Stein operator of the form in Section 3,

$$\begin{aligned}
\mathcal{A} g(x^{(1)}, x^{(2)}) &= \frac{1}{2} \left\{ \mathcal{T}^{(1)} g_{x^{(2)}}(x^{(1)}) + \mathcal{T}^{(2)} g_{x^{(1)}}(x^{(2)}) \right\} \\
&= \frac{1}{2} \left\{ (g_{x^{(2)}})'(x^{(1)}) - \frac{x^{(1)} - \mu^{(1)} - \sigma(x^{(2)} - \mu^{(2)})}{1 - \sigma^2} g_{x^{(2)}}(x^{(1)}) \right. \\
&\quad \left. + (g_{x^{(1)}})'(x^{(2)}) - \frac{x^{(2)} - \mu^{(2)} - \sigma(x^{(1)} - \mu^{(1)})}{1 - \sigma^2} g_{x^{(1)}}(x^{(2)}) \right\} \\
&= \frac{1}{2} \left\{ \partial_1 g(x^{(1)}, x^{(2)}) - \frac{x^{(1)} - \mu^{(1)} - \sigma(x^{(2)} - \mu^{(2)})}{1 - \sigma^2} g_{x^{(2)}}(x^{(1)}) \right. \\
&\quad \left. + \partial_2 g(x^{(1)}, x^{(2)}) - \frac{x^{(2)} - \mu^{(2)} - \sigma(x^{(1)} - \mu^{(1)})}{1 - \sigma^2} g_{x^{(1)}}(x^{(2)}) \right\} \\
&= \frac{1}{2} \left\{ \nabla \times g(x^{(1)}, x^{(2)}) - \Sigma^{-1} (x^{(1)} - \mu^{(1)}, x^{(2)} - \mu^{(2)})^T g(x^{(1)}, x^{(2)}) \right\}
\end{aligned}$$

where ∂_i denotes the derivative with respect to $x^{(i)}$. Thus, we recover the score operator given in Eq. 2.

E Energy-based models and score matching

An Energy-based model (EBM) has density given by

$$q(x) = \frac{1}{Z} \exp\{-E(x)\}, \tag{31}$$

where Z is the (generally) intractable normalisation constant (or partition function) and $E(x)$ is called the energy function. EBMs [LeCun et al., 2006] have been used in machine learning contexts for modelling and learning deep generative models. In particular, learning and training EBMs has been studied in machine learning, see for example Song and Kingma [2021]. One of the most popular and relatively stable training objectives is the score-matching (SM) objective given in Eq. (10) [Hyvärinen, 2005],

$$J(p \| q) = \mathbb{E}_p \left[(\log p(x)' - \log q(x)')^2 \right].$$

For an EBM, the SM objective only requires computing $\nabla E(x)$ and $\nabla \cdot \nabla E(x)$ (or $\Delta E(x)$), but it does not require computing the partition function Z. Moreover, by learning the SM objective, we can obtain $\nabla \log q(x)$ directly, to construct an approximate Stein operator. We use this method for the univariate score function estimation.

F More on kernel-based hypothesis tests

F.1 Maximum-mean-discrepancy tests

Maximum-mean-discrepancy (MMD) has been introduced as a kernel-based method to tackle two-sample problems [Gretton et al., 2007], utilising the rich representation of the functions in a reproducing kernel Hilbert space (RKHS) via a kernel mean embedding. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be the kernel associated with RKHS $\mathcal{H}$. The kernel *mean embedding* of a distribution p induced by k is defined as the function

$$\mu_p := \mathbb{E}_{x\sim p}[k(x, \cdot)] \in \mathcal{H}, \tag{32}$$

whenever μ_p exist. The kernel mean embedding in Eq.32 can be estimated empirically from independent and identically distributed (i.i.d.) samples. Given $x_1, \ldots, x_n \sim p$:

$$\widehat{\mu}_p := \frac{1}{n}\sum_{i=1}^{n} k(x_i, \cdot) \tag{33}$$

replacing p by its empirical counterpart $\widehat{p} = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$ where δ_{x_i} denotes the Dirac measure at $x_i \in \mathcal{X}$. For i.i.d. samples, the empirical mean embedding $\widehat{\mu}_p$ is a $\sqrt{n}$-consistent estimator for μ_p in RKHS norm [Tolstikhin et al., 2017], and with n denoting the number of samples, $\|\mu_p - \widehat{\mu}_p\|_{\mathcal{H}} = O_p(n^{-\frac{1}{2}})$. When the sample size n is small, the estimation error may not be negligible.

The MMD between two distributions p and q is defined as

$$\begin{aligned}\mathrm{MMD}(p\|q; \mathcal{H}) &= \sup_{\|f\|_{\mathcal{H}}\leq 1} \mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{\tilde{x}\sim q}[f(\tilde{x})] \\ &= \sup_{\|f\|_{\mathcal{H}}\leq 1} \langle f, \mu_p - \mu_q\rangle_{\mathcal{H}} = \|\mu_p - \mu_q\|_{\mathcal{H}}.\end{aligned} \tag{34}$$

One desirable property for MMD is to be able to distinguish distributions in the sense that $\mathrm{MMD}(p\|q; \mathcal{H}) = 0 \iff p = q$[11]. This property can be achieved via *characteristic kernels* [Sriperumbudur et al., 2011].

It is often more convenient to work with the squared version of MMD:

$$\begin{aligned}\mathrm{MMD}^2(p\|q; \mathcal{H}) &= \|\mu_p - \mu_q\|^2_{\mathcal{H}} = \langle \mu_p, \mu_p\rangle + \langle \mu_q, \mu_q\rangle - 2\langle \mu_p, \mu_q\rangle \\ &= \mathbb{E}_{x,\tilde{x}\sim p} k(x, \tilde{x}) + \mathbb{E}_{y,\tilde{y}\sim q} k(y, \tilde{y}) - \mathbb{E}_{x\sim p, y\sim q} k(x, y).\end{aligned} \tag{35}$$

Given two sets of i.i.d. samples $\mathbb{S}_p = \{x_1, \ldots, x_n\} \overset{i.i.d.}{\sim} p$ and $\mathbb{S}_q = \{y_1, \ldots, y_l\} \overset{i.i.d.}{\sim} q$, an unbiased estimator of Eq.35, based on the empirical estimate of kernel mean embedding in Eq.33, is given by

$$\mathrm{MMD}^2_u(\mathbb{S}_p\|\mathbb{S}_q; \mathcal{H}) = \frac{1}{n(n-1)}\sum_{i\neq i'} k(x_i, x_{i'}) + \frac{1}{l(l-1)}\sum_{j\neq j'} k(y_j, y_{j'}) - \frac{2}{nl}\sum_{ij} k(x_i, y_j). \tag{36}$$

A two-sample test (or two-sample problem) aims to test the null hypothesis $H_0 : p = q$ against the alternative hypothesis $H_1 : p \neq q$. It has been shown that the *asymptotic* distribution of the n-scaled statistic $n \cdot \mathrm{MMD}^2_u(\mathbb{S}_p\|\mathbb{S}_q; \mathcal{H})$ under the null ($p = q$) is that of an infinite weighted sum of χ^2-distribution [Gretton et al., 2012a, Theorem 12], while under the alternative ($p \neq q$), the $\sqrt{n}$-scaled statistic $\sqrt{n} \cdot \mathrm{MMD}^2_u(\mathbb{S}_p\|\mathbb{S}_q; \mathcal{H})$ is asymptotically normally distributed with the mean centered at $\mathrm{MMD}(p\|q; \mathcal{H}) > 0$. Thus, $n \cdot \mathrm{MMD}^2_u(\mathbb{S}_p\|\mathbb{S}_q; \mathcal{H})$ is taken as a test statistic to be compared against the *rejection threshold*. If the test statistic exceeds the rejection threshold, the empirical estimation of the MMD statistic is thought to exhibit significant departure from the null hypothesis so that H_0 is rejected.

As the null distribution is given by an infinite weighted sum of χ^2 random variables which does not have a closed form expression, the null distribution is often simulated via a permutation procedure [Gretton et al., 2008]: Combine and order two sets of samples as $z_i = x_i, i \in [n]$ and $z_j = y_{j-n}, n+1 \leq j \leq n+l$. Let $\mu : [n+l] \to [n+l]$ be a permutation, and write $z^\mu = \{z_{\mu_{(1)}}, \ldots z_{\mu_{(n+l)}}\}$.

[11] Note that MMD is symmetric with respect to p, q, while KSD is not symmetric with respect to p, q.

Sample size N	20	50	100	200	500	1000
MMD	0.08	0.06	**0.36**	**0.9**	**1.00**	**1.00**
MMDAgg	0.06	0.07	0.02	0.03	0.02	0.05
KSD	0.07	0.04	0.04	0.02	0.08	0.06
NP-KSD	0.06	0.03	0.04	0.06	0.05	0.04

Table 5: Type-I error with increasing sample size N for 100 tests. H_0 is Standard Gaussian with $m = 3$; $n = 50$; the test level is $\alpha = 0.05$; $n_{sim} = 500$.Bold values show uncontrolled type-I errors.

Then z^μ is re-split into $\mathbb{S}_p^\mu = \{z_i\}_{1\leq i\leq n}$ and $\mathbb{S}_q^\mu = \{z_j\}_{n+1\leq j\leq n+l}$. The permuted MMD is computed via Eq. 36 as

$$\mathrm{MMD}_u^2(z^\mu) = \mathrm{MMD}_u^2(\mathbb{S}_p^\mu \| \mathbb{S}_q^\mu; \mathcal{H}). \tag{37}$$

For $\mu_1, \ldots, \mu_B$, we obtain $\mathrm{MMD}_u^2(z^{\mu_1}), \ldots, \mathrm{MMD}_u^2(z^{\mu_B})$ and use these values to compute the empirical quantile of the test statistics $\mathrm{MMD}_u^2(\mathbb{S}_p \| \mathbb{S}_q; \mathcal{H})$.

To test whether the implicit generative model can generate samples following the same distribution as the observed sample, it is natural to consider the two-sample problem described above, which tests whether two sets of samples are from the same distribution. In the model assessment context, one set of samples (of size N) are generated from the implicit model, while the other set of samples (of size n) are observed.

The MMD test often assumes that the sample sizes n and l are equal; its asymptotic theoretical guarantees including consistency are valid under the regime that $n, l \to \infty$ [Gretton et al., 2009, 2012b, Jitkrittum et al., 2016]; also the relative model comparisons in Jitkrittum et al. [2018] only considered the cases $n = l$. In our setting, the sample size l is usually denoted by N. For our model assessment problem setting, when n is fixed and $N \to \infty$ is allowed to be asymptotically large, i.e. $n \ll N$, we find that the type-I error may not be controlled. Hence it is not always the case that MMD is able to pick up the distributional difference between two sets of samples under the null hypothesis. A simple experiment in Table 5 shows an example in which the type-I error is not controlled when N is increasing. Hence, MMD is *not* used as comparison for NP-KSD.

The high type-I error of the MMD statistic as N increases, shown in Table 5 can be heuristically explained as follows. Let $\{x_1, \ldots, x_n\}, \{\tilde{x}_1, \ldots, \tilde{x}_N\} \overset{i.i.d.}{\sim} p$ where the two sets of samples are generated from the same distribution. Let $\widehat{\mu}_{p,n} = \frac{1}{n}\sum_{i\in[n]} k(x_i, \cdot)$, and $\widehat{\mu}_{p,N} = \frac{1}{N}\sum_{j\in[N]} k(\tilde{x}_j, \cdot)$. The empirical MMD between $\widehat{\mu}_{p,n}$ and $\widehat{\mu}_{p,N}$ can be seen as

$$\|\widehat{\mu}_{p,n} - \widehat{\mu}_{p,N}\|_{\mathcal{H}}^2 = \|(\widehat{\mu}_{p,n} - \mu_p) - (\widehat{\mu}_{p,N} - \mu_p)\|_{\mathcal{H}}^2, \tag{38}$$

where MMD aims to detect the asymptotic equality of $(\widehat{\mu}_{p,n} - \mu_p)$ and $(\widehat{\mu}_{p,N} - \mu_p)$. When n is small and fixed, and $n \ll N$, the difference is non-trivial and a rich-enough kernel is able detect this difference, leading to MMD rejecting the null hypothesis although it is true.

MMDAgg, a non-asymptotic MMD-based test Recently, Schrab et al. [2021] proposed an aggregated MMD test that can incorporate the setting $n \neq N$ as long as there exists a constant $C > 0$ such that $n \leq N \leq cn$. Under this condition, MMDAgg is a consistent non-asymptotic test with controlled type-I error, see Table 5. We use it as the competitor method in the main text and in Table 5; in Table 5 KSD is also included, as preferred method when the underlying null distribution is known.

The MMDAgg test statistic is computed by aggregating a set of MMD statistic based on different choices of kernel, e.g. Gaussian kernels with different bandwidth. MMDAgg takes into account a number of choices of bandwidth $\lambda \in \Lambda$ where Λ is a finite set. Let $\widehat{M}_\lambda$ denote the empirical MMD using a kernel with bandwidth λ. Each λ is weighted, via w_λ, where $\sum_{\lambda\in\Lambda} w_\lambda = 1$. In Schrab et al. [2021] (as well as in our implemented experiments), uniform weights are chosen; $w_\lambda \equiv w = \frac{1}{|\Lambda|}$. Denote by B_1 the number of samples used to simulate the null distribution for quantile estimation [12] and denote by B_2 the number of simulated samples used to estimate the empirical rejection probability.

[12]This is the same as notion B in the main text as well as in Eq. 37.

Define $\widehat{q}^{B_1}_{\lambda,1-\alpha}(z^{B_1})$ as the conditional empirical $(1-\alpha)$-quantile when MMD uses a kernel with bandwidth λ, estimated from the permutation procedure with B_1 permutations using Eq.37. Then for a fixed test level α, a quantity u is estimated via the bi-section method such that

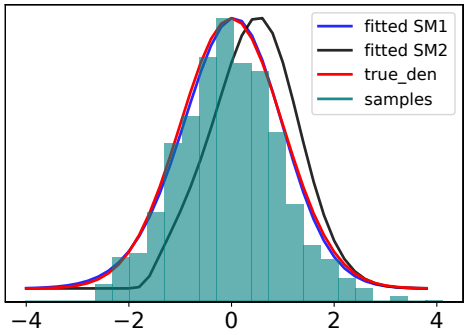
$$\mathbb{P}\left(\max_{\lambda\in\Lambda}(\widehat{M}_\lambda - \widehat{q}^{B_1}_{\lambda,uw}(z^{B_1})) > 0\right) \leq \alpha. \tag{39}$$

We reject H_0 if for any $\lambda \in \Lambda$ and with the estimated $\widehat{u}$, we have that $\widehat{M}_\lambda$ exceeds the rejection probability in Eq. 39; otherwise we do not reject H_0. In this way, MMDAgg does not only achieve the desired non-asymptotic type-I error but is also able to explore a wide range of kernels in order to produce stronger test power. The estimation and adjustment in Eq. 39 is absent in standard MMD testing procedures Gretton et al. [2012a] and is crucial for the MMDAgg being a non-asymptotic test.

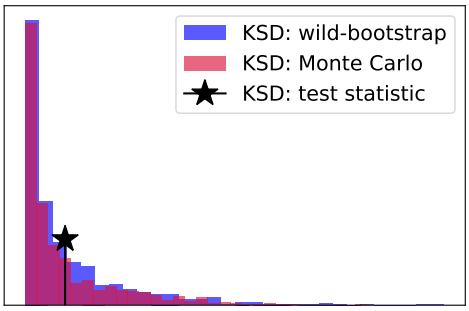
F.2 Wild-bootstrap on KSD testing procedures

In Chwialkowski et al. [2014] a wild bootstrap procedure is proposed to simulate the null distribution via so-called wild-bootstrap samples. For KSD, Chwialkowski et al. [2014] have shown weak asymptotic convergence to the null distribution for deterministic and bounded kernels. For the NP-KSD test statistic, wild-bootstrap samples do not necessarily converge to the null distribution, due to the estimation difference $(\widehat{s}^t - s)$, creating a random Stein kernel for NP-KSD. Perhaps therefore unsurprisingly, the wild bootstrap procedure NP-KSD does not control the type-I error correctly. Instead, we consider a Monte Carlo procedure to simulate the null distribution of NP-KSD. While Monte Carlo estimation is more computationally intensive than wild-bootstrap, it is an accurate method by design.

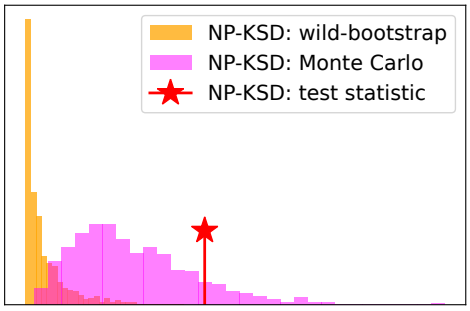
Figure 5 illustrates this point. Figure 5(a) shows samples from a Gaussian distribution. The true density is plotted in red. Two score matching density estimates, SM1 and SM2, are calculated; SM1 presents a good fit whereas SM2 is a less accurate estimate. For KSD, which is applicable when the underlying null distribution is known, Figure 5(b) shows that the Monte Carlo distribution and the wild-bootstrap distribution are close and reach the same conclusion for the KSD test statistic. Using the well-fitting SM1 score density estimate, Figure 5(c) gives the Monte Carlo distribution and the wild-bootstrap distribution. The wild-bootstrap distribution is close to the wild-bootstrap distribution for the KSD. In contrast, it differs considerably from the Monte Carlo distribution and would reject the null hypothesis although it is true. Figure 5(d) shows a similar result for the not so well fitting estimate SM2. The wild-bootstrap distribution is now more spread out but the observed test statistic is still in the tail of this distribution, whereas it is close to the center of the Monte Carlo distribution. In the synthetic experiments for the MoG model in the main text, the model misspecification **NP-KSD_G** falls under this setting. These plots illustrate that using wild-bootstrap samples in this scenario could lead to erroneous conclusions. Hence we use Monte Carlo samples.



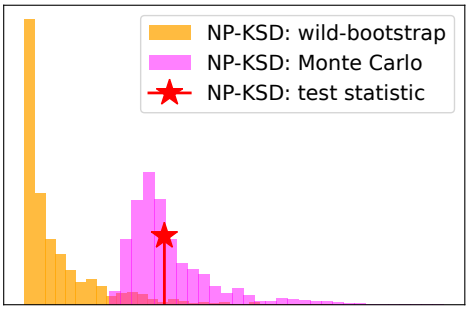
(a) Samples and fitted densities



(b) Simulated null distributions from KSD



(c) Simulated null distributions from fitted SM1



(d) Simulated null distributions from fit SM2

Figure 5: Visualisation for NP-KSD and KSD testing procedures. For KSD, the wild-bootstrap distribution roughly agrees with the Monte Carlo distribution, whereas for NP-KSD, the wild-bootstrap distribution deviates strongly from the Monte Carlo distribution, indicating a danger of reaching an erroneous conclusion when using wild-bootstrap samples in this scenario.