A Kernelised Stein Statistic for Assessing Implicit Generative Models

Anonymous Author(s)
Affiliation
Address
email

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 imbalance datasets. Training deep generative models has been widely studied in the 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 training quality has been studied relying largely on the specific choice of training loss, which does not directly translate into a measure of sample

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

quality; in the case of the log-likelihood [Theis et al., 2015]. Common quality assessment measures for implicit generative models, on images for example, include Inception Scores (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 [2017] proposed to assess sample quality using discrepancy measures called *kernelised Stein discrepancy* (KSD). 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 provides a solution of this problem.

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. 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 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 we introduce the NP-KSD in Section 3 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. Code is also attached in the supplementary material.

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 [Barbour and Chen, 2005, Barbour, 2005, Barbour et al., 2018]. 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 utilises the score-function, see for example Mijoule et al. [2021]. The gradient operator is denoted by ∇ and taken to be a column vector. The *score function* of q is defined as $\mathbf{s}_q = \nabla \log q = \frac{\nabla q}{q}$ (with the convention that $\mathbf{s}_q \equiv 0$ outside of Ω_q). Let $\mathbf{f} = (f_1, \dots, f_m)^\top$ where $f_i : \mathbb{R}^m \to \mathbb{R}, \forall i$, are differentiable. 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 Stein operators can be used to assess discrepancies between two probability distributions; the Stein discrepancy between probability distribution p and q (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 considered, resulting in the *kernel Stein discrepancy* (KSD) defined as [Gorham and Mackey, 2017]

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

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, e.g. V-statistic [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 studied as discrepancy measure between distributions for testing model goodness-of-fit [Chwialkowski et al., 2016, Liu et al., 2016].

KSD testing procedure Suppose we have observed samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ from the *unknown* distribution p. To test the null hypothesis $\mathrm{H}_0 : p = q$ against the (broad class of) alternative hypothesis $\mathrm{H}_1 : p \neq q$, 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 for Stein operator in Eq. 2, which is based on an explicit parametric density, is no longer feasible.

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

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 uni-variate 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 uni-variate conditional score functions.

We denote observed data $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ with $\boldsymbol{z}_i = (z_i^{(1)}, \ldots, z_i^{(m)})^\top \in \mathbb{R}^m$; and denoting the generative model as 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 uni-variate 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 $w_i = \frac{1}{m}$, 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 uni-variate 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 are considered 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 has been considered [Wang et al., 2018] and [Zhuo et al., 2018] in the context of graphical models [Liu and Wang, 2016]. 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.

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: $\widehat{s}_{t,N}^{(i)}(z^{(i)}|t(z^{(-i)})), \forall i \in [m]$.

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],

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

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 uni-variate score function $s^{(i)}$. Let $\mathcal{T}^{(i)}$ be a Stein operator for the uni-variate differentiable probability distribution $Q^{(i)}$ of the generalised density operator form Eq. 8. Let*

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

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. Inspired by Xu and Reinert [2021] and Xu and Reinert [2022], we use low-dimensional measurable non-trivial summary statistics 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 uni-variate 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 uni-variate or multi-variate, and they may attempt to capture useful distributional features. Here we consider uni-variate summary statistics such as the sample mean.

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

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

Non-parametric kernel Stein discrepancy With the well-defined non-parametric Stein operator, we define the corresponding non-parametric Stein discrepancy (NP-KSD) using the Stein operator in Eq. 9, the Stein discrepancy notion in Eq. 3 and choosing as set of test functions the unit ball of the RKHS within unit ball RKHS. Similarly to Eq. 4, we define the NP-KSD with summary statistic t as

$$\text{NP-KSD}_t(G\|p) = \sup_{f \in \mathcal{B}_1(\mathcal{H})} \mathbb{E}_p[\widehat{\mathcal{A}}_{t,N}^{B} f]. \tag{11}$$

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

$$\text{NP-KSD}_t^2(G\|p) = \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\widehat{u}_{t,N}^B(\boldsymbol{x},\tilde{\boldsymbol{x}})], \tag{12}$$

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

$$\widehat{\text{NP-KSD}}_t^2(G\|p) = \frac{1}{n^2}\sum_{i,j\in[n]}[\widehat{u}_{t,N}^B(\boldsymbol{z}_i,\boldsymbol{z}_j)], \tag{13}$$

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

$$\text{KSD}_t^2(q_t\|p) = \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p}[\langle \mathcal{A}_t k(\boldsymbol{x},\cdot), \mathcal{A}_t k(\tilde{\boldsymbol{x}},\cdot)\rangle_{\mathcal{H}} \quad \text{with} \quad \mathcal{A}_t g(\boldsymbol{x}) := \frac{1}{m}\sum_{i=1}^m \mathcal{T}_{q_t}^{(i)} g_{x^{(-i)}}(x^{(i)}) \tag{14}$$

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

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

The proof of Theorem 3.2, which is found in App. B, also shows that the distribution $\text{NP-KSD}_t^2(G\|p) - \text{KSD}_t^2(q_t\|p)$ involves 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 $\mathbb{S} = \{\boldsymbol{z}_1,\dots,\boldsymbol{z}_n\}$, we aim to test the null hypothesis $H_0 : \mathbb{S} \sim G$ versus the alternative $H_1 : \mathbb{S} \not\sim G$. This test assumes that samples generated from G follows some (unknown) distribution q and $\mathbb{S}$ 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 [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.

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.

[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 $\mathbb{S} = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_n\}$; 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 $\{i_1, \dots, i_B\}$ from $[m]$, with replacement.
3: Compute $\tau = \widehat{\text{NP-KSD}}^2(\widehat{s}_t; \mathbb{S})$ in Eq. (13).
4: Simulate $\mathbb{S}_i = \{\boldsymbol{y}'_1, \dots, \boldsymbol{y}'_n\}$ 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 $\{\tau_1, \dots, \tau_b\}$.

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

Related works To assess whether an implicit generative models can generate samples that are *significantly* good for the desired data model, several hypothesis testing procedures have been studied. Jitkrittum et al. [2018] has proposed kernel-based test statistics, Relative Unbiased Mean Embedding (Rel-UME) test and Relative Finite-Set Stein Discrepancy (Rel-FSSD) test for relative model goodness-of-fit, i.e. whether model S is a better fit than model R. While Rel-UME is applicable for implicit generative models, Rel-FSSD still requires explicit knowledge of the unnormalised density. The idea for assessing sample quality for implicit generative models is through addressing two-sample problem, where samples generated from the implicit model are compared with the observed data. In this sense, maximum-mean-discrepancy (MMD) may also apply for assessing sample qualities for the implicit models. With 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, as the distribution is represented via samples, the two-sample based assessment suffers from limited probabilistic information from the implicit model and low estimation accuracy when the sample size for observed data is small.

4 Experiments

4.1 Baseline and competing approaches

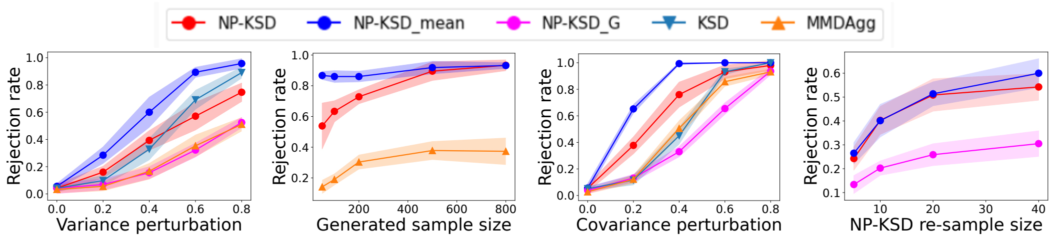
We illustrate the proposed NP-KSD testing procedure with different choice of summary statistics. We denote by **NP-KSD** the version which uses the estimation of the conditional score, i.e. $t(x^{(-i)}) = x^{(-i)}$; by **NP-KSD_mean** the version which uses conditioning on the mean statistics, i.e. $t(x^{(-i)}) = \frac{1}{m-1}\sum_{j\neq i} x^{(j)}$; 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 for more details. 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, then identifiability of q_t through a KSD method may not be guaranteed. Details while MMD is not included in this list are found in Appendix F.

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

[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.



(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).

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 rate against this perturbation of covariance terms are presented in Figure 1(c). The results show consistent 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), while the variability of the average test power also 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 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, which essentially acts as a generator of the null distribution.

MNIST Dataset This dataset contains 28×28 grey-scale images of handwritten digits [LeCun et al., 1998][4]. It consist of $60,000$ training samples and $10,000$ test samples. Deep gen-

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

erative 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. 600 samples are generated from the generative models and 100 samples are used for the test; test level $\alpha = 0.05$. From Table 1, we see that all the deep generative models have high rejection rate, showing that the trained models are not good enough. 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.

	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.

CIFAR10 Dataset This dataset contains 32×32 RGB coloured images [Krizhevsky, 2009][5]. 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][6], which is created to differ from CIFAR10 to investigate generalisation power for training classifiers. 800 samples are generated from the generative models and 200 samples are used for the test; test level $\alpha = 0.05$. 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.

	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.

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 in the case of medical applications could be fatal, and in the case of loan applications for example could be detrimental to personal or business development. NP-KSD provides a method for assessing synthetic data generators 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 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. Yet NP-KSD is an important step towards understanding black-box data generating methods and thus understanding their potential shortcomings.

[5] `https://pytorch.org/vision/stable/generated/torchvision.datasets.CIFAR10.html`
[6] `https://github.com/modestyachts/CIFAR-10.1/tree/master/datasets`

References

AD Barbour. Multivariate Poisson–binomial approximation using Stein's method. In *Stein's Method And Applications*, pages 131–142. World Scientific, 2005.

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

AD Barbour, Malwina J Luczak, and Aihua Xia. Multivariate approximation in total variation, ii: Discrete normal approximation. *The Annals of Probability*, 46(3):1405–1440, 2018.

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.

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, 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, 2012.

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 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.

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.

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.

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.

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

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.

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? [No]
   (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]