Sliced Kernelized Stein Discrepancy

Wenbo Gong
University of Cambridge
wg242@cam.ac.uk

Yingzhen Li
Microsoft Research Cambridge
Yingzhen.Li@microsoft.com

José Miguel Hernández-Lobato
University of Cambridge
jmh233@cam.ac.uk

Abstract

Kernelized Stein discrepancy (KSD), though being extensively used in goodness-of-fit tests and model learning, suffers from the curse-of-dimensionality. We address this issue by proposing the sliced Stein discrepancy and its scalable and kernelized variants, which employs kernel-based test functions defined on the optimal one-dimensional projections instead of the full input in high dimensions. When applied to goodness-of-fit tests, extensive experiments show the proposed discrepancy significantly outperforms KSD and various baselines in high dimensions. For model learning, we show its advantages by training an independent component analysis when compared with existing Stein discrepancy baselines. We further propose a novel particle inference method called sliced Stein variational gradient descent (S-SVGD) which alleviates the mode-collapse issue of SVGD in training variational autoencoders.

1 Introduction

Discrepancy measures are important in statistics and machine learning, as they can be used to quantify differences between two probability distributions. Among many existing discrepancy measures, Stein discrepancy (SD) is unique in that it only requires samples from one distribution and the score function (i.e. the gradient up to a multiplicative constant) from the other one [17]. SD is an integral probability metric (IPM) [46] and its computation requires finding an optimal test function within a given function family. However, this optimization can be made analytic by using a reproducing kernel Hilbert space (RKHS) as the test function family, resulting in the kernelized Stein discrepancy (KSD) [37, 11]. They have been widely used in both Goodness-of-fit (GOF) tests [37, 27, 23, 11, 28] and model learning [35, 18, 22, 40, 36, 38, 16, 51].

Although theoretically elegant, KSD, especially with RBF kernel, suffers from the “curse-of-dimensionality” issue, which leads to significant deterioration of test power in GOF tests [11, 23, 43] and mode collapse in particle inference [51]. A few attempts have been made at addressing this problem, however, they either are limited to specific applications [31, 8] or require significant approximations in practice [43] and consequently lacks guarantees for GOF tests. As an alternative, in this work we adopt the idea of “slicing” to address the curse-of-dimensionality issue of KSD, especially with the RBF kernel. In a nutshell, the key idea is to project the score function and test inputs onto slicing directions. This allows us to define an IPM that only requires one-dimensional kernels for describing the test functions. Specifically, our contributions are as follows.

- We propose a novel family of discrepancies called sliced Stein discrepancy (SSD). We further derive a scalable variant called max sliced kernelized Stein discrepancy (maxSKSD) using kernel tricks and the optimal test direction. Both discrepancies are theoretically validated in terms of their correctness.

Preprint. Under review.
We derive a GOF test using an unbiased estimator of maxSKSD given the optimal test directions and provide an analysis of its asymptotic behaviour. It is empirically evaluated on benchmark problems based on Gaussian distributions and restricted Boltzmann machine models \cite{17, 20, 21}. Compared with previous SD baselines, maxSKSD is shown to be superior in terms of both the robustness to high dimensions and the statistical power.

We evaluate the usefulness of maxSKSD in model learning by two schemes. First, we train an independent component analysis (ICA) \cite{22, 23, 24} in high dimensions by directly minimising maxSKSD, which results in faster convergence and better performance. Further, we propose a particle inference algorithm called the sliced Stein variational gradient descent (S-SVGD) as a novel variant of the SVGD \cite{40}. We show that S-SVGD alleviates the posterior mode collapse of SVGD when applied to training variational autoencoders \cite{30, 41}.

2 Background

2.1 Kernelized Stein Discrepancy

For two probability distributions \( p \) and \( q \) supported on \( \mathcal{X} \subseteq \mathbb{R}^D \) with continuous differentiable densities \( p(x) \) and \( q(x) \), we define the score \( s_p(x) = \nabla_x \log p(x) \) and \( s_q(x) \) accordingly. For a test function \( f : \mathcal{X} \to \mathbb{R}^D \), the Stein operator is defined as

\[
A_p f(x) = s_p(x)^T f(x) + \nabla_x^T f(x). 
\]  

(1)

For a function \( f_0 : \mathbb{R}^D \to \mathbb{R} \), the Stein class \( \mathcal{F}_q \) of \( q \) is defined as the set of functions satisfying the Stein identity \cite{42}: \( \mathbb{E}_q[s_q(x)f_0(x) + \nabla_x f_0(x)] = 0 \). This definition can be generalized to a vector function \( f : \mathbb{R}^D \to \mathbb{R}^D \) as well, and we overload the notation to write \( f \in \mathcal{F}_q \) if \( f_i \) belongs to the Stein class of \( q \) for each \( i \in D \). Then the Stein discrepancy \cite{37, 17} is defined as

\[
D(q, p) = \sup_{f \in \mathcal{F}_q} \mathbb{E}_q[A_p f(x)] = \sup_{f \in \mathcal{F}_q} \mathbb{E}_q[(s_p(x) - s_q(x))^T f(x)]. 
\]  

(2)

When \( \mathcal{F}_q \) is sufficiently rich, and \( q \) vanishes at the boundary of \( \mathcal{X} \) the supremum is obtained at \( f^*(x) = s_p(x) - s_q(x) \) with some mild regularity conditions on \( f \). This describes the score difference at location \( x \) \cite{22}. Thus, the Stein discrepancy is 0 iff. the score of \( p \) equals the score of \( q \) and \( p = q \) a.e.. To obtain an analytic form of SD, the kernelized Stein discrepancy (KSD) \cite{37, 11} restricts the test functions to be in a \( D \)-dimensional RKHS \( \mathcal{H}_D \) with kernel \( k \) such that for a function \( f : \mathcal{X} \to \mathbb{R}^D \), \( f(x)_i = \langle f_i, k(x, \cdot) \rangle_{\mathcal{H}} \) and \( \|f\|_{\mathcal{H}_D}^2 = \sum_i \|f_i\|_{\mathcal{H}_D}^2 \). By defining

\[
u_p(x, x') = s_p(x)^T s_p(x') k(x, x') + s_p(x)^T \nabla_x k(x, x') + s_p(x')^T \nabla_{x'} k(x, x') + \text{Tr}(\nabla_{x', x'} k(x, x'))
\]

the analytic form of KSD is the following:

\[
D^2(q, p) = \left( \sup_{f \in \mathcal{H}_D, \|f\|_{\mathcal{H}_D} \leq 1} \mathbb{E}_q[A_p f(x)] \right)^2 = \mathbb{E}_{q(x)q(x')}[\nu_p(x, x')]. 
\]  

(3)

2.2 Stein Variational Gradient Descent

Although SD and KSD can be directly minimized for variational inference (VI) \cite{40, 42, 43}. Liu and Wang \cite{40} proposed a novel particle inference algorithm based on KSD called Stein variational gradient descent (SVGD) as an alternative. SVGD applies a sequence of deterministic transformations to a set of points such that each of these mappings maximally decreases the Kullback-Leibler (KL) divergence from the particles’ underlying distribution \( q \) to the target \( p \).

To be specific, we define the mapping \( T(x) : \mathbb{R}^D \to \mathbb{R}^D \) as \( T(x) = x + \epsilon \phi(x) \) where \( \phi \) is a smooth function that characterises the perturbations. The following result from \cite{40} shows the connections between SD and the derivative of the KL divergence.

**Lemma 1.** \cite{40} Let \( T(x) = x + \epsilon \phi(x) \) and \( q[T](z) \) be the density of \( z = T(x) \) when \( x \sim q(x) \).

With \( A_p \) the Stein operator defined in Eq.\((1)\), we have

\[
\nabla_x KL[q[T]||p]|_{\epsilon=0} = -\mathbb{E}_q[A_p \phi(x)].
\]  

(4)
We propose the sliced Stein discrepancy (SSD).

We verify the proposed SSD is a valid discrepancy measure, namely, we apply maxSKSD to GOF tests, and develop two ways for model learning.

Theorem 1. (SSD Validity) If assumptions 1-4 in appendix A are satisfied, then for two probability distributions \( p \) and \( q \) over \( \mathbb{R}^D \),

\[
\phi^{*}_{p,q}(\cdot) = \mathbb{E}_q[\nabla_x \log p(x)k(x, \cdot) + \nabla_x k(x, \cdot)]
\]

and \( \nabla_x K \mathbb{L}[q||p]|_{x=0} = -D^2(q,p) \).

The first term in Eq. (5) is called drift, which drives the particles towards a mode of \( p \). The second term controls the repulsive force, which spreads the particles around the mode. When particles stop moving, the KL decrease magnitude \( \epsilon D^2(q,p) \) is 0, which means the KSD is zero and \( p = q \) a.e.

### 3 Sliced Kernelized Stein Discrepancy

We propose the sliced Stein discrepancy (SSD) and its scalable and kernelized version named maxSKSD. Theoretically, we prove their correctness as discrepancy measures. Methodology-wise, we apply maxSKSD to GOF tests, and develop two ways for model learning.

#### 3.1 Sliced Stein Discrepancy

The curse of dimensionality issue of the Stein discrepancy (Eq. (2)) comes from two sources: the score function \( s_p(x) \) and the test function \( f(x) \) defined on \( \mathcal{X} \subseteq \mathbb{R}^D \). We address the first source by projecting the score of \( p \) onto a slicing direction \( r \), i.e. \( s^r_p(x) = s_p(x)^T r \). This corresponds to slicing the \( p \) distribution through the \( r \) direction at location \( x \). One can show that \( p = q \) a.e. iff. \( s^r_p(x) = s^r_q(x) \) for all \( r \). As \( s^r_p(x) \) is unavailable, we can attempt to verify this by using test functions similar to the Stein discrepancy verifying the score differences (Eq. (2)). However, to address the second source mentioned above, one should avoid using the high dimensional \( x \) as the input to the test functions. A first attempt would be to re-use \( r \) to project \( x \) so that the input is \( x^T r \). However the optimal test function for the projected score is \( f^*(x) \propto (s^r_p(x) - s^r_q(x)) \) (similar to the SD in Section 2.1). Thus, using \( x^T r \) for \( f \) could create information loss. The key idea to avoid this is to introduce an infinite number of additional test directions \( g \) and evaluate the test functions on \( x^T g \). The intuition of the proposed discrepancy is further explained in appendix B.1.

In detail, assume two distributions \( p \) and \( q \) supported on \( \mathbb{R}^D \) with differentiable densities \( p(x) \) and \( q(x) \), and define the test functions \( f(\cdot; r, g) : \mathbb{R}^D \rightarrow \mathbb{R} \) such that \( f(x; r, g) = f_{rg}(x^T g) \). The proposed sliced Stein discrepancy (SSD), defined using two uniform distributions \( p_r(r) \) and \( p_g(g) \) over the hypersphere \( S^{D-1} \), is given by

\[
S(q,p) = \mathbb{E}_{p_r, p_g} \left[ \sup_{f_{rg} \in \mathcal{F}_q} \mathbb{E}_q[s^r_p(x)f_{rg}(x^T g) + r^T g \nabla_{x^T g} f_{rg}(x^T g)] \right], \tag{6}
\]

where \( f_{rg} \in \mathcal{F}_q \) means \( f(\cdot; r, g) \in \mathcal{F}_q \) and \( \mathcal{F}_q \) represents the Stein class of \( q \). Using slicing and defining the test functions in projected spaces indeed reduces the dimensions from \( \mathbb{R}^D \) to \( \mathbb{R}^\ell \) when compared to the SD, see Eq. (6). The information loss caused by the projections \( x^T g \) is addressed by averaging over an infinite number of \( g \) from \( p_g \).

We verify the proposed SSD is a valid discrepancy measure, namely, \( S(q,p) = 0 \) iff. \( q = p \) a.e..

Theorem 1. (SSD Validity) If assumptions 1-4 in appendix A are satisfied, then for two probability distributions \( p \) and \( q \), \( S(q,p) \geq 0 \), and \( S(q,p) = 0 \) if and only if \( p = q \) a.e.

Despite this attractive theoretical result, SSD is difficult to compute in practice. Specifically, the expectations over \( r \) and \( g \) can be approximated by Monte Carlo but this typically requires a very large number of samples in high dimensional problems [14]. This limitation can be addressed by relaxing the requirement of infinite number of projections. In particular, it suffices to use only a finite number of slicing directions \( r \) from an orthogonal basis \( O_r \) of \( \mathbb{R}^D \), e.g. the standard basis of one-hot vectors, and the corresponding optimal test direction \( g_r \) for each \( r \). We name such version of SSD as maxSSD, which are defined as follows and validated in Corollary 1.1.

\[
S_{\max}(q,p) = \sum_{r \in O_r} \sup_{f_{rg} \in \mathcal{F}_q, g \in S^{D-1}} \mathbb{E}_q[s^r_p(x)f_{rg}(x^T g) + r^T g \nabla_{x^T g} f_{rg}(x^T g)]. \tag{7}
\]
Corollary 1.1. \((\text{maxSSD})\) Assume that the conditions in Theorem 3 are satisfied, then \(S_{\text{max}}(q, p) = 0\) if and only if \(p = q\) a.e.

3.2 Closed form Solution with the Kernel Trick

The optimal test function for the direction pair \(r\) and \(g\) is intractable without further assumptions on the test function families. This introduces another scalability issue as optimizing these test functions explicitly can be time consuming. To tackle this issue, we apply the kernel trick to obtain an analytic solution for the optimal test functions. Assume for each test function \(f_{rg} \in H_{rg}\) where \(H_{rg}\) is a scalar-valued RKHS equipped with kernel \(k(x, x'; r, g) = k_{rg}(x^T g, x'^T g)\) that satisfies assumption 5 in appendix A and \(f_{rg}(x^T g) = \langle f_{rg}, k_{rg}(x^T g, \cdot) \rangle_{H_{rg}}\). We define the following quantities:

\[
\begin{align*}
\xi_{p, r, g}(x, \cdot) &= s^T_p(x)k_{rg}(x^T g, \cdot) + r^T g\nabla x^T g k_{rg}(x^T g, \cdot), \\
h_{p, r, g}(x, y) &= s^T_p(x)k_{rg}(x^T g, y^T g)\phi_s(y) + r^T g\phi_s(y)\nabla x^T g k_{rg}(x^T g, y^T g) + \quad (r^T g)^2\nabla x^T g, y^T g + k_{rg}(x^T g, y^T g).
\end{align*}
\]

The following theorem describes the optimal test function inside SSD (Eq.(6)) and maxSSD (Eq.(7)).

**Theorem 2.** \((\text{Closed form solution})\) If \(\mathbb{E}_q[h_{p, r, g}(x, x)] < \infty\), then

\[
D^2_{rg}(q, p) = \left\| \sup_{f_{rg} \in H_{rg}, ||f_{rg}|| \leq 1} \mathbb{E}_q[s^T_p(x)f_{rg}(x^T g) + r^T g\nabla x^T g f_{rg}(x^T g)] \right\|^2
= \left\| \mathbb{E}_q[\xi_{p, r, g}(x)] \right\|^2_{H_{rg}} = \mathbb{E}_q(x|q(x')| h_{p, r, g}(x, x')).
\]

Next, we propose the kernelized version of SSD with orthogonal slicing basis \(O_r\), called SKSD, and show it is a valid discrepancy.

**Theorem 3.** \((\text{SKSD as a discrepancy})\) For two probability distributions \(p\) and \(q\), given assumptions 1, 2 and 5 in appendix A, and \(\mathbb{E}_q[h_{p, r, g}(x, x)] < \infty\) for all \(r\) and \(g\), we define SKSD as

\[
SK_o(q, p) = \sum_{r \in O_r} \int_{\mathbb{S}^{d-1}} p_gD^2_{rg}(q, p)dg,
\]

which is equal to 0 if and only if \(p = q\) a.e.

Note that, instead of the integration over \(g\), we only need to use the optimal \(g_r\) for each \(r \in O_r\), resulting in a matrix \(G \in \mathbb{R}^{D \times D}\) obtained by optimization. We name this discrepancy as maxSKSD, or maxSKSD-\(g\) when we need to distinguish it from another variant described later.

**Corollary 1.** \((\text{maxSKSD})\) Assume the conditions in Theorem 3 are satisfied. Then

\[
SK_{\text{max}}(q, p) = \sum_{r \in O_r} \sup_{g_r} D^2_{rg}(q, p)
\]

is equal to 0 if and only if \(p = q\) a.e.

One can also use the optimal \(r\) to replace the summation over \(O_r\), which is later shown to be beneficial in GOF tests. We call this discrepancy \(\text{maxSKSD-rg}\), and its validity can be proved accordingly.

**Kernel choice** In the experiments we use the RBF kernel with median heuristics, which satisfies the conditions in Theorem 3. However, better kernels, e.g. deep kernels which evaluate a given kernel function on the transformed input \(\phi(x)\), might be preferred for structured data such as images. Using deep kernels for KSD is straight-forward and it only requires the deep kernel to be characteristic. But a naive application of deep kernels to maxSKSD would result in a kernel evaluated on \(\phi(x^T g)\), which is less desirable. Instead, we propose an adapted form of Eq.(11) using the projected features \(\phi(x)^T g\), and validate the approach by assuming the mapping is smooth and injective. We include the details in appendix B and leave the experiments for future work.

3.3 Application of maxSKSD

**Goodness-of-fit Test** Assume the optimal test directions \(g_r \in G\) are available, and in practice gradient based optimization can be used, maxSKSD (Eq.(12)) can be approximated using U-statistics. Given i.i.d. samples \(\{x_i\}_{i=1}^N \sim q\), we have an unbiased minimum variance estimator:

\[
\hat{SK}_{\text{max}}(q, p) = \frac{1}{N(N-1)} \sum_{r \in O_r} \sum_{1 \leq i < j \leq N} h_{p, r, g_r}(x_i, x_j).
\]
The asymptotic behavior of the estimator is analyzed in appendix E.1. We use bootstrap [27, 24, 2] to determine the threshold for rejecting the null hypothesis, see Algorithm 1 in appendix E.1.

**Model Learning**  The proposed maxSKSD can be applied to model learning in two ways. First, it can be directly used as a training objective for machine learning models, in such case the $q$ is the data distribution and $p$ is the model to be learned. The second model learning scheme is to leverage the particle inference for latent variables and train the model parameters using an EM-like [12] algorithm. For this, maxSKSD can be used to improve a well-known particle inference called SVGD [36], and we name the improved algorithm sliced-SVGD (SSVGD). In short, we define a specific form of the perturbation as $\phi(x) = [\phi_{g_1}(x^T g_1), \ldots, \phi_{g_D}(x^T g_D)]^T$ and modify the proofs of Lemmas 1 & 2 accordingly. The resulting S-SVGD algorithm uses kernels defined on one dimensional projected samples, which sidesteps the vanishing repulsive force problem of SVGD in high dimensions. This problem leads to an under-estimation of uncertainty in the target distribution. We illustrate this in Figure 1 by estimating the variance of a standard Gaussian with the particles obtained by SVGD or S-SVGD (see appendix C.3). We see that as the dimension increases, SVGD severely under-estimates the variance of $p$, while the S-SVGD remains robust. Furthermore, the validity of S-SVGD is justified since in such case the KL gradient equals to maxSKSD which is a valid discrepancy. Readers are referred to appendix E.2 for the derivations.

### 4 Experiments

#### 4.1 Goodness of fit test

We evaluate maxSKSD (Eq. (12)) for GOF tests in high dimensional problems. First, we demonstrate its robustness to the increasing dimensionality using the Gaussian GOF benchmarks [27, 23, 11]. Next, we show our method is also a more powerful GOF test by using 50 dimensional Restricted Boltzmann Machine (RBM) [37, 23, 27, 18]. In appendix F.3, we further apply the maxSKSD to select the step size for stochastic gradient Hamiltonian Monte Carlo (SGHMC) [9]. The baselines in comparison include GOF test methods using Gaussian or Cauchy random Fourier features (RFF) [39], KSD with RBF kernel [37, 11], finite set Stein discrepancy (FFSD) with random or optimized test locations [27], random feature Stein discrepancy (RFSD) with L2 SechExp and L1 IMQ kernels [23], and maximum mean discrepancy (MMD) [19] with RBF kernel. Notice that we use gradient descent to obtain the test directions $g_i$ (and potentially the slicing directions $r$) for Eq. (12), this is different from the approaches that finds the hyper-parameters with maximized test power [27, 49, 54]. For fair comparisons, we do not train any hyper-parameters in the following except for FFSD with optimized test locations.

**4.1.1 GOF Tests with High dimensional Gaussian Benchmarks**

We conduct 4 different benchmark tests with $p = \mathcal{N}(0, I)$: (1) **Null test**: $q = p$; (2) **Laplace**: $q(x) = \prod_{d=1}^{D} \text{Lap}(x_d | 0, 1/\sqrt{2})$ with matched mean and variance to $p$; (3) **Multivariate-t**: $q$ is multivariate-t with 5 degrees of freedom; (4) **Diffusion**: $q(x) = \mathcal{N}(0, \Sigma_1)$ where the $1^{\text{st}}$ entry in the diagonal of $\Sigma_1$ is 0.3 and the rest is the same as in $I$.

**Setup** We set the significance level $\alpha = 0.05$ for all GOF experiments. For FFSD and RFSD, we use the open-sourced code from the original publications. We only consider maxSKSD-g here as it already achieves nearly optimally. For more details on the setup, we refer to appendix F.1.

Figure 2 shows the GOF test performances and the corresponding discrepancy values. In summary, the proposed maxSKSD outperforms the baselines in all tests, where the result is robust to the increasing dimensions and the discrepancy values match the expected behaviours.

**Null** In this setting, the null rejection rate is expected to be closed to the significance level. The left-most column in Figure 2 shows that all methods behave as expected, except for RFSD with L2 SechExp kernel. All the discrepancy values oscillate around 0, with the KSD being less stable.
Figure 2: Each column reports GOF test results for a different alternative hypothesis, with the upper panel showing the rejection rate of the Null hypothesis and the lower panel showing the discrepancy value averaged over all trials. Both quantities are plotted w.r.t. the number of dimensions.

Laplace and Multivariate-t The two middle columns of Figure 2 show that maxSKSD-g achieves a nearly perfect rejection rate consistently as the dimension increases, while the test power for all baselines decreases significantly. For the discrepancy values, similar to the KL divergence between $q$ and $p$, maxSKSD-g linearly increases with dimensions due to the independence assumptions.

Diffusion This is a more challenging setting since $p$ and $q$ only differ in one of their marginal distributions, which becomes increasingly difficult to identify as the dimension increases. As shown in the rightmost column of Figure 2, all methods failed in high dimensions except maxSKSD-g, which still consistently achieves optimal performance. Regarding the discrepancy values, we expect a positive constant due to the one marginal difference between $p$ and $q$. Only maxSKSD-g behaves as expected as the problem dimension increases. The decreasing value at the beginning is probably due to the difficulty in finding the optimal direction $g$ in high dimensions when the training set is small.

4.1.2 RBM GOF test

We demonstrate the power of the GOF test with maxSKSD. The baselines are the same as before, but we now also include results for maxSKSD-rg.

Setup The test setup is similar to the RBM in [37, 27, 23] where different amounts of noise are injected into the weight to form the alternative hypothesis $q$. The samples are drawn using block Gibbs samplers. Refer to appendix F.2 for details.

Figure 3 shows that maxSKSD based methods dominate the baselines, especially with maxSKSD-rg significantly outperforming the others. At perturbation level 0.01, maxSKSD-rg achieves 0.96 rejection rate, while the rejection rates for the other methods are all below 0.5. This ablation result demonstrates the advantages of optimizing the slicing directions $r$.

4.2 Model Learning

We evaluate the efficiency of maxSKSD-based algorithms in training machine learning models. First, we use independent component analysis (ICA) [20, 26, 7] which is often used as a benchmark for evaluating training methods for energy-based model [18, 20, 26, 7]. Our approach trains the ICA model by directly minimizing maxSKSD. Next, we evaluate the proposed S-SVGD particle inference algorithm, when combined with amortization [16, 38], in the problem of training a variational autoencoder (VAE) [30, 41] on binarized MNIST. Appendix G.5 also shows results for S-SVGD when training a Bayesian neural network (BNN) on UCI datasets [15].
| Method | Dimension |
|--------|-----------|
|        | $D = 10$ | $D = 20$ | $D = 40$ | $D = 60$ | $D = 80$ | $D = 100$ | $D = 200$ |
| KSD    | -10.23   | -15.98  | -34.50  | -56.87  | -86.09  | -116.51 | -329.49   |
| LSD    | -10.42   | -14.54  | -17.16  | -15.05  | -12.39  | -5.49   | 46.63     |
| maxSKSD| -10.45   | -14.50  | -17.28  | -15.70  | -11.91  | -4.21   | 47.72     |

Table 2: Average log likelihood on first 5,000 test images for different $D$ of latent dimensions.

| Method       | Latent Dim |
|--------------|------------|
|              | $D = 16$   | $D = 32$ | $D = 48$ | $D = 64$ |
| Vanilla VAE  | -91.50     | -90.39   | -90.58   | -91.50  |
| SVGD VAE     | **-88.58** | -90.43   | -93.47   | -94.88  |
| S-SVGD VAE   | -89.17     | **-87.55** | **-87.74** | **-87.78** |

Table 3: Label entropy and accuracy for imputed images.

| Method       | Entropy | Accuracy |
|--------------|---------|----------|
| Vanilla VAE  | 0.297   | 0.718    |
| SVGD VAE     | 0.538   | 0.691    |
| S-SVGD VAE   | **0.542** | **0.728** |

4.2.1 ICA

ICA consists of a simple generative process $z \sim \text{Lap}(0, 1)$ and $x = Wz$, where the model parameters are a non-singular matrix $W \in \mathbb{R}^{D \times D}$. The log density for $x$ is $\log p(x) = \log p_z(W^{-1}x) + C$, where the normalization constant $C$ can be ignored when training with Stein discrepancies. We train the models on data sampled from a randomly initialized ICA model and evaluate the corresponding test log likelihoods. We compare maxSKSD with KSD and the state-of-the-art LSD [18], which outperforms sliced score matching [45], noise-contrastive estimation (NCE) [20] and conditional NCE [7]. For more details on the setup, we refer the reader to appendix G.2.

Table 1 shows average test log likelihoods for the different training methods. Both maxSKSD and LSD are robust to increasing values of $D$, with maxSKSD being better when $D$ is very large. Also at $D = 200$, maxSKSD converges significantly faster than LSD (see Figure 6 in appendix G.3). We argue that this faster convergence is due to the closed-form solution for the optimal test functions, whereas LSD requires adversarial training. While KSD is also kernel-based, this method suffers from the curse-of-dimensionality and it fails to train the model properly for $D > 20$. Further, it starts to diverge when $D = 200$. These results show that the proposed maxSKSD can successfully address the problems of KSD with high dimensional data.

4.2.2 Amortized SVGD

Finally, we consider training VAEs with implicit encoders on dynamically binarized MNIST. The decoder is trained similarly as in vanilla VAEs, but the encoder is trained by amortization [16, 38], which minimizes the mean square error between the initial samples from the encoder, and the modified samples driven by the SVGD/S-SVGD dynamics (Algorithm 3 in appendix G.4).

Setup We report performance in terms of test log-likelihood (LL). Furthermore we consider an imputation task, by removing the pixels in the lower half of the image and imputing the missing values using (approximate) posterior sampling from the VAE models. The performance is measured in terms of imputation diversity and correctness, using label entropy and accuracy. Note that the original amortized SVGD approach [16] requires manual tuning of the coefficient for the repulsive force to counter mode collapse, which is unnecessary in our case. For more details setup, we refer to appendix G.4.

Table 2 reports the average test LL results, and we observe that S-SVGD is much more robust to the increasing dimensionality compared to SVGD. To be specific, with $D = 16$ latent space, SVGD performs the best and is 2.92 nats better than vanilla VAE. S-SVGD performs slightly worse than SVGD (0.59 nats). However, when the dimension starts to increase, LL of SVGD drops significantly, and for $D = 64$, which is a common choice for training VAE on MNIST, it performs significantly worse than vanilla VAE. On the other hand, S-SVGD dominates SVGD and Vanilla VAE. Notice that the purpose of this experiment is to show the robustness of S-SVGD instead of achieving the state-of-the-art performance. Still the performance can be easily boosted, e.g. running longer S-SVGD steps before encoder update, we leave it for the future work.
For the imputation task, we compute the label entropy and accuracy for the imputed images (Table 3). We observe S-SVGD has higher label entropy compared to vanilla VAE and better accuracy compared to SVGD. This means both S-SVGD and SVGD capture the multi-modality nature of the posterior compared to uni-modal Gaussian distribution. However, high label entropy itself may not be a good indicator for the quality of the learned posterior. One can think of a counter-example that the imputed images are diverse but does not look like any digits. This may also gives a high label entropy but the quality of the posterior is poor. Thus, we use the accuracy to indicate the “correctness” of the imputed images, with higher label accuracy meaning the imputed images are close to the original image. Together, a good model should give a higher label entropy along with the high label accuracy. We observe S-SVGD has more diverse imputed images with high imputation accuracy.

5 Related Work

Stein Discrepancy  SD [17] and KSD [57, 11] are originally proposed for GOF tests. Since then research progress has been made to improve these two discrepancies. For SD, LSD [18, 22] is proposed to increase the capacity of test functions using neural networks with $L_2$ regularisation. On the other hand, FSSD [27] and RFSD [23] aim to reduce the computation cost of KSD from $O(n^2)$ to $O(n)$ where $n$ is the number of samples. Still the curse-of-dimensionality issue remains to be addressed in KSD, and the only attempt so far (to the best of our knowledge) is the kernelized complete conditional Stein discrepancy (KCC-SD [43]), which share our idea of avoiding kernel evaluations on high dimensional inputs. Specifically, KCC-SD compares the conditional distributions $p(x_i|\bar{x})$ and $q(x_i|\bar{x})$ using Stein’s method for all indices $i$, which indeed reduces the input dimensions from $D$ to 1. However, KCC-SD requires sampling from $q(x_i|\bar{x})$ which is often intractable, therefore significant approximations towards $q(x_i|\bar{x})$ are needed. This makes KCC-SD less well-suited for GOF test due to the difficulty of asymptotic analysis. On the other hand, our approach does not require this approximation, and the corresponding estimator is well-behaved asymptotically.

Sliced Wasserstein Distance and Score matching  The “slicing” idea has been widely used in statistics and machine learning, and the methods related to our approach include sliced Wasserstein distance (SWD) [31] and sliced score matching (SSM) [45, 44]. However, these approaches use slicing to address the computational inefficiencies rather than statistical difficulties in high dimensions. To be specific, Wasserstein distance, though computationally intractable in high dimensions, has analytic solutions in one dimension. This is exploited by SWD to compute the Wasserstein distance on one-dimensional projection of the data then integrate over all directions. Some other variants of SWD are also proposed to address the inefficiency of projections in high dimensions [14] and relax the constraints of linear projection [32]. For SSM, the purpose of slicing is to avoid the computation of the trace of Hessian, which is also known as Hutchson’s trick [25].

Particle Inference  Despite the neat form of SVGD, it has been shown to suffer from the mode collapse problem in high dimensions. Zhuo et al. [51] proposed message passing SVGD to tackle this problem by defining local kernels using the Markov blanket of the variables in the graphical model. However, this algorithm can only be applied in certain problems due to the requirement of the specific graph structure. Projected SVGD (pSVGD) is a very recent attempt [8] which projects and updates the high dimension particles in an adaptively constructed low dimensional space, resulting in a biased inference algorithm. Compare to pSVGD, the major difference here is that S-SVGD still updates the particles in the original space while computing kernels on 1-dimensional projections. Furthermore, S-SVGD can theoretically recover the correct target distribution since maxSKSD is a valid discrepancy. There is no real-world experiments provided in [8], and a stable implementation of pSVGD is non-trivial, so we did not consider pSVGD when selecting the baselines.

6 Conclusion

We proposed sliced Stein discrepancy (SSD), as well as its scalable and kernelized version maxSKSD, to address the curse-of-dimensionality issues in Stein discrepancy. The key idea is to project the score function on one-dimensional slices and define (kernel-based) test functions on one-dimensional projections. We also theoretically prove their validity as a discrepancy measure. We conduct extensive experiments including GOF tests and model learning to show maxSKSD’s improved performance and robustness in high dimensions. There are three exciting avenues of future research. First, although validated by our theoretical study in appendix D, practical approaches to incorporate deep kernels into SSD remains an open question. Second, the performance of maxSKSD crucially depends on the
optimal projection direction, so better optimization methods to efficiently construct this direction is needed. Lastly, we believe “slicing” is a promising direction for kernel design to increase the robustness to high dimensional problems in general. For example, MMD can be easily extended to high dimensional two-sample tests using this kernel design trick.

**Broader Impact**

This work proposes a new family of discrepancy measures that target at the curse-of-dimensionality issue of KSD. First, due to its property, the proposed discrepancy can have potential impacts at training and evaluating deep generative models, especially for problems in high dimensions. For example, in chemistry and physics, our proposed method can be used to evaluate the good-of-fit of a particular model to certain data for model comparisons or train the model to approximate desired distributions (e.g. Boltzmann distributions in statistical mechanics). In addition, it can also be impactful in areas using MCMC based methods. Specifically, assessing the quality of MCMC samples remains an open challenge especially in high dimensions. Our proposed method is a promising candidate for such problem.

**Acknowledgments and Disclosure of Funding**

Wenbo Gong is supported by CSC-Cambridge Trust Scholarship.

**References**

[1] Michael Arbel, Dougal Sutherland, Mikolaj Binkowski, and Arthur Gretton. On gradient regularizers for mmd gans. In *Advances in Neural Information Processing Systems*, pages 6700–6710, 2018.

[2] Miguel A Arcones and Evarist Gine. On the bootstrap of u and v statistics. *The Annals of Statistics*, pages 655–674, 1992.

[3] Adi Ben-Israel. The change-of-variables formula using matrix volume. *SIAM Journal on Matrix Analysis and Applications*, 21(1):300–312, 1999.

[4] Mikolaj Binkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[5] Ronald N Bracewell. Strip integration in radio astronomy. *Australian Journal of Physics*, 9(2):198–217, 1956.

[6] Claudio Carmeli, Ernesto De Vito, Alessandro Toigo, and Veronica Umanitá. Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61, 2010.

[7] Ciwan Ceylan and Michael U Gutmann. Conditional noise-contrastive estimation of unnormalised models. *arXiv preprint arXiv:1806.03664*, 2018.

[8] Peng Chen and Omar Ghattas. Projected Stein variational gradient descent. *arXiv preprint arXiv:2002.03469*, 2020.

[9] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691, 2014.

[10] Andreas Christmann and Ingo Steinwart. Support vector machines.

[11] Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit. JMLR: Workshop and Conference Proceedings, 2016.

[12] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[13] Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the sliced Wasserstein distance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3483–3491, 2018.
[14] Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. Max-sliced Wasserstein distance and its use for gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 10648–10656, 2019.

[15] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[16] Yihao Feng, Dilin Wang, and Qiang Liu. Learning to draw samples with amortized Stein variational gradient descent. arXiv preprint arXiv:1707.06626, 2017.

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

[18] Will Grathwohl, Kuan-Chieh Wang, Jorn-Henrik Jacobsen, David Duvenaud, and Richard Zemel. Cutting out the middle-man: Training and evaluating energy-based models without sampling. arXiv preprint arXiv:2002.05616, 2020.

[19] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. Journal of Machine Learning Research, 13(Mar):723–773, 2012.

[20] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 297–304, 2010.

[21] Wassily Hoeffding. A class of statistics with asymptotically normal distribution. In Breakthroughs in Statistics, pages 308–334. Springer, 1992.

[22] Tianyang Hu, Zixiang Chen, Hanxi Sun, Jincheng Bai, Mao Ye, and Guang Cheng. Stein neural sampler. arXiv preprint arXiv:1810.03545, 2018.

[23] Jonathan Huggins and Lester Mackey. Random feature Stein discrepancies. In Advances in Neural Information Processing Systems, pages 1899–1909, 2018.

[24] Marie Huskova and Paul Janssen. Consistency of the generalized bootstrap for degenerate u-statistics. The Annals of Statistics, pages 1811–1823, 1993.

[25] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. Communications in Statistics-Simulation and Computation, 19(2):433–450, 1990.

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

[27] 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.

[28] Heishiro Kanagawa, Wittawat Jitkrittum, Lester Mackey, Kenji Fukumizu, and Arthur Gretton. A kernel Stein test for comparing latent variable models. arXiv preprint arXiv:1907.00586, 2019.

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

[30] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114, 2013.

[31] Soheil Kolouri, Yang Zou, and Gustavo K Rohde. Sliced Wasserstein kernels for probability distributions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5258–5267, 2016.

[32] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced Wasserstein distances. In Advances in Neural Information Processing Systems, pages 261–272, 2019.

[33] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In Advances in Neural Information Processing Systems, pages 2203–2213, 2017.

[34] Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and DJ Sutherland. Learning deep kernels for non-parametric two-sample tests. arXiv preprint arXiv:2002.09116, 2020.
[35] Qiang Liu and Yihao Feng. Two methods for wild variational inference. *arXiv preprint arXiv:1612.00081*, 2016.

[36] 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.

[37] 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.

[38] Yuchen Pu, Zhe Gan, Ricardo Henao, Chunyuan Li, Shaobo Han, and Lawrence Carin. VAE learning via Stein variational gradient descent. In *Advances in Neural Information Processing Systems*, pages 4326–4345, 2017.

[39] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[40] Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator variational inference. In *Advances in Neural Information Processing Systems*, pages 496–504, 2016.

[41] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[42] Robert J Serfling. *Approximation theorems of mathematical statistics*, volume 162. John Wiley & Sons, 2009.

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

[44] 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.

[45] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. *arXiv preprint arXiv:1905.07088*, 2019.

[46] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. On integral probability metrics, $\phi$-divergences and binary classification. *arXiv preprint arXiv:0901.2698*, 2009.

[47] Charles Stein, Persi Diaconis, Susan Holmes, Gesine Reinert, et al. Use of exchangeable pairs in the analysis of simulations. In *Stein’s Method*, pages 1–25. Institute of Mathematical Statistics, 2004.

[48] Charles Stein et al. 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.

[49] Dougal J Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alex Smola, and Arthur Gretton. Generative models and model criticism via optimized maximum mean discrepancy. *arXiv preprint arXiv:1611.04488*, 2016.

[50] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016.

[51] Jingwei Zhuo, Chang Liu, Jiaxin Shi, Jun Zhu, Ning Chen, and Bo Zhang. Message passing Stein variational gradient descent. *arXiv preprint arXiv:1711.04425*, 2017.
A Definitions and Assumptions

**Definition A.1.** *(Stein Class [37]*) Assume distribution \( q \) has continuous and differentiable density \( q(x) \). A function \( f \) defined on the domain \( X \subset \mathbb{R}^D \), \( f : X \to \mathbb{R} \) is in the Stein class of \( q \) if \( f \) is smooth and satisfies
\[
\int_X \nabla_x (f(x) q(x)) \, dx = 0 \tag{14}
\]

We can easily see that the above holds true for \( X = \mathbb{R}^D \) if
\[
\lim_{||x|| \to \infty} q(x) f(x) = 0 \tag{15}
\]

This can be verified using integration by parts or divergence theorem. Specifically, if \( q(x) \) vanishes at infinity, then it only requires the test function \( f \) to be bounded. This definition can be generalized to a vector valued function \( f : X \to \mathbb{R}^D \). We say such function \( f \) is in Stein class of \( q \) if the member of \( f \), \( f_i \), belongs to the Stein class of \( q \) for all \( i \in D \).

**Definition A.2.** *(Stein Class [37]*) A kernel \( k(x, x') \) is said to be in the Stein class of \( q \) if \( k(x, x') \) has continuous second order partial derivatives, and both \( k(x, \cdot) \) and \( k(\cdot, x) \) are in the Stein class of \( q \) for any fixed \( x \).

**Radon Transform** In machine learning literature, Radon transform has been used as the primary tool to derive sliced Wasserstein distance [32, 13, 14]. To be specific, the standard Radon transform, denoted as \( R \), is a map from \( L^1 \) integrable functions \( I \in L^1(\mathbb{R}^D) \) to the infinite set of its integrals over the hyperplane of \( \mathbb{R}^D \). Specifically, for \( L^1 \) integrable functions:
\[
L^1(\mathbb{R}^D) = \{ I : \mathbb{R}^D \to \mathbb{R} \mid \int_{\mathbb{R}^D} |I(x)| \, dx < \infty \},
\]

the Radon transform is defined by
\[
R[I](l, g) = \int_{\mathbb{R}^D} I(x) \delta(l - (x, g)) \, dx \tag{17}
\]

for \( (l, g) \in \mathbb{R} \times S^{D-1} \) where \( S^{D-1} \subset \mathbb{R}^D \) stands for a unit sphere in \( \mathbb{R}^D \). For fixed \( g \), this defines a continuous function \( R[I](\cdot, g) : \mathbb{R} \to \mathbb{R} \) which is the projection of function \( I \) on to the hyper-plane with its normal vector defined by \( g \) and offset defined by \( l \).

In the following we state the assumptions that we used to prove our main results.

**Assumption 1** *(Properties of densities)* Assume the two probability distributions \( p, q \) has continuous differentiable density \( p(x), q(x) \) supported on \( \mathbb{R}^D \). Density \( q \) satisfies: \( \lim_{||x|| \to \infty} q(x) = 0 \).

**Assumption 2** *(Regularity of score functions)* Denote the score function of \( p(x) \) as \( s_p(x) = \nabla_x \log p(x) \in \mathbb{R}^D \) and score function of \( q(x) \) accordingly. Assume the score functions satisfy
\[
\int_{\mathbb{R}^D} q(x) \|(s_p(x) - s_q(x))^T r\| \, dx < \infty \tag{18}
\]
\[
\int_{\mathbb{R}^D} q(x) \|(s_p(x) - s_q(x))^T r\|^2 \, dx < \infty
\]

for all \( r \) where \( r \in \mathbb{R}^D \) is a vector sampled from a uniform distribution over a unit ball \( S^{D-1} \). In other words, the score difference, when projected on the \( r \) direction, is both \( L^1 \) and \( L^2 \) integrable with respect to the probability measure defined by \( q(x) \, dx \). These conditions are used to ensure both the Radon transform and the proposed divergence are well defined.

**Assumption 3** *(Stein Class of test functions)* Assume the test function \( f(\cdot, r, g) : \mathbb{R}^D \to \mathbb{R} \) is smooth and belongs to the Stein class of \( q \).

**Assumption 4** *(Bounded Radon transformed functions)* Define
\[
I_{q,p} = q(x) (s_p(x) - s_q(x))^T r \tag{19}
\]

We assume the Radon transformation of \( I_{q,p} \), \( R[I_{q,p}](l, g) \) is bounded for all \( g \), where \( g \) is sampled from a uniform distribution over a unit ball \( S^{D-1} \). Namely, \( ||R[I_{q,p}](l, g)||_{\infty} < \infty \).
We provide an explanation on the roles played by $r$ and $g$ where $x$ is necessary to express the projected score difference using a wide range of one-dimensional representations.

To see why using a test direction $g$ is necessary, we provide a counter-example in the case of using one-hot vectors w.l.o.g., as all the orthonormal basis in $\mathbb{R}^D$ are equivalent up to rotations. Now consider two probability distributions $p$ and $q$ supported on $\mathbb{R}^D$, where $p(x) = \prod_i^D p(x_i)$ and $q(x_i) = p(x_i)$, respectively. Importantly, $q$ distribution might not be factorized. Then we have

$$D(q, p) = \sum_i^D \sup_{f_i \in \mathcal{F}_q} \mathbb{E}_q[s_p^i(x_i)f_i(x_i) + \nabla_{x_i} f_i(x_i)] = 0$$  \hspace{1cm} (21)$$

where $s_p^i(x_i) = \nabla_{x_i} \log p(x_i)$ and $q_i = q(x_i)$. The second equality is from Stein identity due to the matching marginal of $p$ and $q$. However, it is not necessary that $p = q$, e.g. each dimensions in $q$ is correlated. The main reason for this counter-example is that the test function only observes the marginal input $x_i$ and ignores any correlations that may exist in $q$.

### B.2 Proof of Theorem

We split the proof of theorem into two parts. First, we prove the ‘if’ part by the following proposition.
Proposition 1. (SSD Detect Convergence) If two distributions $p = q$ a.e., and assumption 1-4 are satisfied, then $S(q,p) = 0$.

Proof. To prove SSD can detect convergence of $q$ and $p$, we first introduce the Stein identity \[47, 37\].

Lemma 3. (Stein Identity) Assume $q$ is a smooth density satisfied assumption 1, then we have

$$E_q[s_q(x)f(x)^T + \nabla f(x)] = 0$$

for any functions $f : \mathbb{R}^D \to \mathbb{R}^D$ in Stein class of $q$.

From the Stein identity, and $p = q$ a.e., we can take the trace of the Stein identity:

$$\int q(x)[s_q(x)^T f(x) + \nabla^T f(x)] dx = 0$$

where $F(x) : \mathbb{R}^D \to \mathbb{R}^D$ and it belongs to the Stein class of $q$.

Next, we choose a special form for $F(x)$. For particular sliced direction pair $r$ and $g$, we define

$$F(x) = \begin{bmatrix} r_1 f_{r,g}(x^T g) \\ r_2 f_{r,g}(x^T g) \\ \vdots \\ r_D f_{r,g}(x^T g) \end{bmatrix}$$

where $r = [r_1, r_2, \ldots, r_D]^T$.

From the assumption 3 and definition of Stein class of $q$ for vector functions in section 2.1, it is trivial that $F(x)$ belongs to the Stein class of $q$. Substitute this $F(x)$ into Stein discrepancy Eq.(2), we have

$$\int q(x)[s_q(x)^T F(x) + \nabla^T F(x)] dx = 0$$

$$\Rightarrow \int q(x)[s_q(x)^T r f_{r,g}(x^T g) + r^T g \nabla x r f_{r,g}(x^T g)] dx = 0$$

for all test functions $f_{r,g}$ that belongs to Stein class of $q$. Therefore, Eq.(6) is 0 if $p = q$ a.e. \qed

The ‘only if’ part of theorem 1 is less direct to prove. Before we start this journey, we need to introduce some properties relating to Radon transform.

Lemma 4. (Fourier Slice Theorem [5]) For a particular smooth function $f(x) : \mathbb{R}^D \to \mathbb{R}$ that satisfies assumptions of Radon transforms, we define $F_D$ as the $D$ dimensional Fourier transform operator, $S_1$ as a slice operator which extracts 1 dimensional central slice of a function and $R$ as the Radon transform operator. Thus, for a slice direction $g$, we have the following equivalence

$$S_1[F_D[f]](\omega, g) = F_1[R[f](l, g)](\omega).$$

(22)

This theorem implies the following two operations are equivalent.

- First apply $D$ dimensional Fourier transform to a function $f$ and then take a slice that goes through the origin with direction $g$ from the transformed function.
- First apply the Radon transform with direction $g$ to the function $f$ and then apply one dimensional Fourier transform to the projected function.

Next we show some properties related to the rotated distributions.

Lemma 5. (Marginalization Invariance of rotated distribution) Assume we have a probability distribution $q$ supported on $\mathbb{R}^D$, a rotation matrix $G \in \mathbb{R}^{D \times D}$ and a test function $f : \mathbb{R}^D \to \mathbb{R}$, we can define the corresponding rotated distribution $q_G$ after applying the rotation matrix $G$. Thus, we have the following identity

$$\int q_G(x)f(G^{-1}x)dx = \int q(x)f(x)dx.$$  

(23)
Proof. By the definition of rotation and change of variable formula, we define \( y = Gx \), we can show
\[
q_G(y) = q(x)|G^{-1}|
= q(G^{-1}y) \times 1
= q(G^{-1}y)
\]
where \(|G^{-1}|\) represents the determinant of the inverse rotation matrix. Thus, by change of variable formula, we have
\[
\int q_G(y)f(G^{-1}y)dy = \int q(x)f(x)|G|dx
= \int q(y)f(y)dy.
\]

This identity is useful when dealing with the rotated distributions. Next, we introduce the generalization of change-of-variable formula, which is often used in differential geometry.

**Lemma 6.** (Change of Variable Formula using Matrix Volume [3]) If \( U \) and \( V \) are sets in spaces with different dimensions, say \( U \in \mathbb{R}^n \) and \( V \in \mathbb{R}^m \) with \( n > m \), and \( \phi : U \rightarrow V \) is a continuously differentiable injective function and \( f : \mathbb{R}^m \rightarrow \mathbb{R} \) is integrable on \( V \), we have the following change of variable formula:
\[
\int_V f(v)dv = \int_U (f \circ \phi)(u)\text{vol}\ J\phi(u)du \tag{24}
\]
where \( \text{vol}\ J\phi(u) \) is the matrix volume of the Jacobian matrix \( J\phi(u) = \partial(v_1, \ldots, v_m)/\partial(u_1, \ldots, u_n) \).

Particularly, if \( J\phi(u) \) is of full column rank, then \( \text{vol}\ J\phi = \sqrt{\det J\phi^T J\phi} \).

Next, we derive the key lemma that establishes the relationship between the conditional expectation of rotated distribution and Radon transform of the original distribution.

**Lemma 7.** (Conditional Expectation = Radon Transform) For a particular test direction \( g_d \in \mathbb{R}^D \), we can define an arbitrary rotation matrix \( G \in \mathbb{R}^{D \times D} \) that the \( d \)-th entry is the test direction \( g_d \). We assume the probability distribution \( q(x) \) is supported on \( \mathbb{R}^D \), and \( x \) is a constant \( x = p \). Further, let define the mapping \( x = Gu \) for \( u \in \mathbb{R}^D \) and \( x \) represents \( x \setminus x_d \) (all elements of \( x \) except \( x_d \)). Thus, with the smooth test function \( f : \mathbb{R}^D \rightarrow \mathbb{R} \) and the assumptions in Radon transformation being true, we have the following identity:
\[
\int_{\mathcal{X}_d} q_G(x_d, x_{-d})f(G^{-1}x)dx_{-d} = \int q(u)f(u)\delta(p-u^T g_d)du \tag{25}
\]
where \( \mathcal{X}_d = \{ x \in \mathbb{R}^D | x_d = p \} \).

Proof. From the definition of \( x = Gu \), we can define the rotation matrix \( G \) as following:
\[
G = \begin{bmatrix}
g_1^T \\
\vdots \\
g_d^T \\
\vdots \\
g_D^T
\end{bmatrix}
\]
where \( g_d = [g_{d,1}, \ldots, g_{d,D}]^T \). Thus, assume \( x_d = p \), we can write down

\[
x = Gu = \begin{bmatrix}
\sum_{k=1}^D g_{1,k} u_k \\
\vdots \\
p \\
\vdots \\
\sum_{k=1}^D g_{D,k} u_k
\end{bmatrix}.
\]

Thus, the Jacobian matrix can be written as

\[
J = \frac{\partial(Gu)}{\partial u} = \begin{bmatrix}
g_{1,1} & \cdots & g_{d-1,1} & g_{d+1,1} & \cdots & g_{D,1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
g_{1,D} & \cdots & g_{d-1,D} & g_{d+1,D} & \cdots & g_{D,D}
\end{bmatrix} = \begin{bmatrix}
g_1 & \cdots & g_{d-1} & g_{d+1} & \cdots & g_D
\end{bmatrix}.
\]

By the definition of rotation matrix, the Jacobian matrix is clearly full column rank. Thus, from Lemma 5, we have

\[
\det J = \prod_{i=1}^D g_{i,i} = \prod_{i=1}^D g_{i,i}.
\]

Consider the inner supreme inside the Eq.(6), by Proposition 1, we have

\[
\int_{\mathcal{X}_d} qG(x_d, x_d) f(G^{-1} x) dx_d = \int_{\mathcal{X}_d} q(G^{-1} x) f(G^{-1} x) \times 1 dx_d
\]

\[
= \int_{\mathcal{U}} q(u) f(u) \delta(p - u^T g_d) du
\]

where \( \mathcal{U} = \{ u \in \mathbb{R}^D \mid g_d^T u = p \} \). Thus, we have

\[
\int_{\mathcal{X}_d} qG(x_d, x_d) f(G^{-1} x) dx_d = \int_{\mathcal{U}} q(u) f(u) \delta(p - u^T g_d) du
\]

Now, we can prove the `only if` part of Theorem 1 using the above lemmas.

Proof. In order to prove equation (Eq.(6)) being 0 implies \( p = q \) a.e., the strategy is to construct a lower bound for Eq.(6) by choosing a particular test function. We also need to make sure this lower bound is greater or equal to 0 and is 0 only if \( p = q \) a.e. Thus, if the Eq.(6) is 0, it implies the lower bound is 0 and \( q = p \) a.e.

Consider the inner supreme inside the Eq.(6), by Proposition 1 we have

\[
\mathbb{E}_q[(s_p(x) - s_q(x))^T r_{f,g}(x^T g)] = \mathbb{E}_q[(s_p(x) - s_q(x))^T r_{f,g}(x^T g)].
\]

Now we apply the Lemma 5 and assume \( y = Gx \) and \( y_d = g^T x \), then, Eq.(26) can be rewritten as

\[
\int q(x) [s_p(x) - s_q(x)]^T r_{f,g}(x^T g) dx
\]

\[
= \int qG(y_d, y_d - d) [\nabla g^{-1} y \log \frac{p(G^{-1} y)}{q(G^{-1} y)}]^T r_{f,g}(y_d) dy_d.
\]
The next step is to choose a specific form for the test function \( f_{r,g}(y_d) \). Define

\[
f_{r,g}(y_d) = \int q_G(y_d, s_d)[\nabla G^{-1} s_d \log \frac{p(G^{-1} s_d)}{q(G^{-1} s_d)}]^T r dy_d.
\]  

(28)

First, we need to make sure this selected test function indeed satisfies assumption 3, namely, it needs to be in the Stein class of \( q \). By Lemma[7], this selected test function can be re-written into

\[
\int q_G(y_d, s_d)[\nabla G^{-1} s_d \log \frac{p(G^{-1} s_d)}{q(G^{-1} s_d)}]^T r dy_d = \int q_G(y_d)[\nabla x \log \frac{p(x)}{q(x)}]^T r \delta(y_d - x^T g)dx
\]

\[= \mathcal{R}[I_{q,p}](y_d, g).
\]

This is exactly the Radon transform of the function \( I_{q,p} = q(x)(s_p(x) - s_q(x))^T r \). By assumption 4, this Radon transform is bounded. Thus, together with assumption 1, we can show this Radon transformed function indeed belongs to the Stein class of \( q \).[4]

Now by substituting this specific test function Eq.(28) into Eq.(27), and defining \( u = [u_1, \ldots, y_d, \ldots, u_D]^T \), we have

\[
\int q_G(y_d, u_d)[\nabla G^{-1} u_d \log \frac{p(G^{-1} u_d)}{q(G^{-1} u_d)}]^T r du_d dy_d = \int q_G(y_d, u_d)[\nabla G^{-1} u_d \log \frac{p(G^{-1} u_d)}{q(G^{-1} u_d)}]^T r du_d dy_d
\]

\[= \int f_{r,g}^1(y_d)dy_d = 1
\]

\[\geq 0.
\]

(29)

Thus we have constructed a lower bound (Eq.(29)) for the supremum in Eq.(6) and it is greater than 0. Next, we show the expectation of this lower bound over \( p_q \) and \( p_r \) is 0 only if \( p = q \) a.e.. If so then Eq.(6) is 0 only if \( p = q \) a.e..

First, it is clearly that \( 1 = 0 \) iff. \( f_{r,g}(y_d) = 0 \) a.e. By Lemma[7], we have \( f_{r,g}(y_d) = \mathcal{R}[I_{q,p}](y_d, g) \). Thus, we have

\[1 = 0 \Rightarrow \mathcal{R}[I_q](y_d, g) = \mathcal{R}[I_p](y_d, g) \text{ a.e.}
\]

where \( I_q = q(x)s_q(x)^T r \) and \( I_p = q(x)s_p(x)^T r \).

Now we define the \( D \) dimensional Fourier transform operator \( F_D \), slice operator \( S_1 \) as in Theorem[4]. Based on Fourier sliced theorem[4], we have

\[\mathcal{R}[I_q](y_d, g) = \mathcal{R}[I_p](y_d, g)
\]

\[\Rightarrow \mathcal{F}_1[I_q](y_d, g) = \mathcal{F}_1[I_p](y_d, g)
\]

\[\Rightarrow S_1[F_D(I_q)](\cdot, g) = S_1[F_D(I_p)](\cdot, g).
\]

(30)

This means the one dimensional slice at direction \( g \) for Fourier transform \( F_D(I_q) \) and \( F_D(I_p) \) are the same. Also note that the discrepancy (Eq.(6)) is defined by integrating over test directions \( q \) with a uniform distribution \( p_q(g) \) over \( \mathbb{S}^{D-1} \). This means if the discrepancy is zero, then Eq.(30) must hold true for \( g \) a.e. over the hyper-sphere. Thus, we can show

\[F_D(I_q) = F_D(I_p) \text{ a.e.}
\]

(31)

It is well-known that the Fourier transform is injective, thus, for any direction \( r \), we have

\[F_D(I_q) = F_D(I_p)
\]

\[\Rightarrow I_q = I_p
\]

\[\Rightarrow q(x)s_q(x)^T r = q(x)s_p(x)^T r
\]

\[\Rightarrow s_q(x)^T r = s_p(x)^T r
\]

(32)

The \( S(q,p) \) (Eq.(5)) also integrates over sliced directions \( r \in \mathbb{S}^{D-1} \), thus, we have

\[s_q(x)^T r = s_p(x)^T r \text{ for all } r \Rightarrow s_q(x) = s_p(x) \Rightarrow p = q \text{ a.e.}
\]

This finishes the proof of the “only if” part: \( S(q,p) \geq 0 \) and is 0 only if \( q = p \) a.e. [10]
B.3 Proof of Corollary 1.1

To prove the corollary 1.1 we first propose a variant of SSD (Eq.(6)) by relaxing the score projection \( r \). We call it orthogonal basis SSD.

**Remark 1.** (Orthogonal basis for SSD) It is not necessary to integrate over all possible \( r \in \mathbb{S}^{D-1} \) for Theorem 1 to hold true. In fact, it suffices to use a set of projections that forms the orthogonal basis \( O_r \in \mathbb{R}^D \). In such case we have

\[
S_o(q, p) = \sum_{r \in O_r} \int_{\mathbb{S}^{D-1}} p_o(g) \sup_{f_r \in \mathcal{F}_q} E_q[s_p^r(x)f_r g(x^T g) + r^T g \nabla \tau g f_r g(x^T g)]dg \tag{33}
\]

is zero if and only if \( p = q \) a.e. One simple choice for \( O_r \) can be \( O_r = \{r_1, \ldots , r_D\} \) where \( r_d \) is one-hot vector with value 1 in \( d \)th component.

To prove Remark 1, we only need to slightly modify the last few steps in the proof of Theorem 1.

**Proof.** We focus on the ‘only if’ part as the other part is trivial. Without loss of generality, we set \( O_r = \{r_1, \ldots , r_D\} \) where \( r_d \) is one-hot vector with value 1 in \( i \)th component. For general \( O_r \), we can simply apply a inverse rotation matrix \( R^{-1} \) to recover this special case.

From Eq.(32), we have for direction \( r_d \),

\[
s_q(x)^T r_d = s_q(x)^T r_d
\]

\[
\Rightarrow \nabla_{x_d} \log q(x_d, x_{-d}) = \nabla_{x_d} \log p(x_d, x_{-d})
\]

\[
\Rightarrow \nabla_{x_d} \log q(x_d, x_{-d}) = \nabla_{x_d} \log p(x_d, x_{-d}).
\]

If the above holds true for all directions \( r_d \in O_r \), then the score of the complete conditional for \( q \) and \( p \) are equal. Then from Lemma 1 in [43], we have \( p = q \) a.e. \( \square \)

Now we can prove Corollary 1.1 using Remark 1.

**Proof.** It is trivial to show \( S_{\text{max}}(q, p) = 0 \) if \( p = q \) a.e. (Stein Identity). Now assume \( S_{\text{max}}(q, p) = 0 \), this means for any direction \( r \in O_r \), and \( g \in \mathbb{S}^{D-1} \), we have

\[
\sup_{f_r \in \mathcal{F}_q} E_q[s_p^r(x)f_r g(x^T g) + r^T g \nabla \tau g f_r g(x^T g)] = 0
\]

This is because we have show in the proof of Theorem 1 that the above term is greater or equal to 0. Then we can directly use Remark 1 to show \( S_{\text{max}}(q, p) = 0 \) only if \( q = p \) a.e. \( \square \)

C SKSD Related Theorems

C.1 Proof of Theorem 2

**Proof.** First, we can verify the following equality using the proof techniques in [37, 11]:

\[
h_{p,r,g}(x, y) = \langle \xi_{p,r,g}(x, \cdot) , \xi_{p,r,g}(y, \cdot) \rangle_{H_{rg}}. \tag{34}
\]

Next, we show that \( \xi_{p,r,g}(x, \cdot) \) is Bochner integrable [10], i.e.

\[
E_q[\xi_{p,r,g}(x)||_{H_{rg}} \leq \sqrt{E_q[\xi_{p,r,g}(x)]^2_{H_{rg}}} = \sqrt{E_q[h_{p,r,g}(x, x)]} \leq \infty. \tag{35}
\]

Thus, we can interchange the expectation and the inner product. Finally we finish the proof by re-writing the supremum in \( S_o(q, p) \): (Eq.(33))

\[
|| \sup_{f_r \in H_{rg}, ||f_r|| \leq 1} E_q[s_p^r(x)f_r g(x^T g) + r^T g \nabla \tau g f_r g(x^T g)]^2 \|
\]

\[
= || \sup_{f_r \in H_{rg}, ||f_r|| \leq 1} E_q[\xi_{p,r,g}(x)||_{H_{rg}}^2 ||f_r||^2_{H_{rg}}] \|
\]

\[
= || \sup_{f_r \in H_{rg}, ||f_r|| \leq 1} (f_r, E_q[s_p^r(x)^T r k_{rg}(x^T g, \cdot) + r^T g \nabla \tau g k_{rg}(x^T g, \cdot), f_r])_{H_{rg}} ||^2 \tag{36}
\]

\[
= ||E_q[h_{p,r,g}(x)||_{H_{rg}}^2 ||_{H_{rg}}
\]

\[
= E_{x, x'} E_q[h_{p,r,g}(x, x')].
\]
C.2 Proof of Theorem 3

Proof. First, we assume $p = q$ a.e. To show $SK_o(q, p) = 0$, we only need to show $D_{rg}^2(q, p) = 0$ for all $r$ and $g$. From Theorem 2 we have

$$D_{rg}^2(q, p) = \langle \mathbb{E}_q[\xi_{p,r,g}(x, \cdot)], \mathbb{E}_q[\xi_{p,r,g}(x', \cdot)] \rangle.$$ 

From Assumption 5, we know $k_{rg}(x^T g, \cdot)$ belongs to the Stein class of $q$. Then we follow the same proof technique in Proposition 2 but replace the test function $f_{rg}(x^T g)$ with $k_{rg}(x^T g, \cdot)$. This gives

$$\mathbb{E}_q[s^r_q(x)^T r k_{rg}(x^T g, \cdot) + r^T g \nabla_{x^T g} k_{rg}(x^T g, \cdot)] = 0,$$

i.e. $\mathbb{E}_q[\xi_{p,r,g}(x, \cdot)] = 0$. Thus, $D_{rg}^2(q, p) = 0$.

Next, we prove that it can detect the non-convergence of $p$ and $q$. We know $SK_o(q, p) = 0$ if and only if $D_{rg}^2(q, p) = 0$. This means

$$D_{rg}(q, p) = 0$$
$$\Rightarrow ||\mathbb{E}_q[\xi_{p,r,g}(x)]||_{H^s} = 0$$
$$\Rightarrow \mathbb{E}_q[\xi_{p,r,g}(x, \cdot)] = 0$$

where the second equality is from theorem 2. From Eq. (37), we can re-write

$$\mathbb{E}_q[\xi_{p,r,g}(x, \cdot)] = \mathbb{E}_q[(s^r_p(x) - s^r_q(x)) k_{rg}(x^T g, \cdot)].$$

Next, we denote $G$ as an arbitrary rotation with the $d$th entry as the test direction $g$, and $y = Gx$ with $y_d = x^T g$. Then from Lemma 5 we have

$$\int q(x) \nabla_x \log \frac{p(x)}{q(x)} r k_{rg}(x^T g, \cdot) dx$$
$$= \int q_G(y_d, y_{-d}) \nabla_{G^{-1}y} \log \frac{q(G^{-1}y)}{p(G^{-1}y)} r k_{rg}(y_d, \cdot) dy_{-d} dy_d$$
$$= \int q_G(y_d) k_{rg}(y_d, \cdot) \int q_G(y_{-d}|y_d) \nabla_{G^{-1}y} \log \frac{q(G^{-1}y)}{p(G^{-1}y)} r dy_{-d} dy_d$$
$$= \int q_G(y_d) k_{rg}(y_d, \cdot) H_r(y_d) dy_d$$

where $H_r(y_d) = \int q_G(y_{-d}|y_d) \nabla_{G^{-1}y} \log \frac{q(G^{-1}y)}{p(G^{-1}y)} r dy_{-d}$. The above equation is exactly the mean embedding of the function $H_r(y_d)$ w.r.t. measure $q_G$. By assumption 5 that the kernel is $C_0$-universal, and by [6], its embedding is zero if and only if $H_r(\cdot) = 0$. This implies

$$H_r(y_d) = \int q_G(y_{-d}|y_d) \nabla_{G^{-1}y} \log \frac{q(G^{-1}y)}{p(G^{-1}y)} r dy_{-d} = 0$$
$$\Rightarrow \int q_G(y_d, y_{-d}) \nabla_{G^{-1}y} \log \frac{q(G^{-1}y)}{p(G^{-1}y)} r dy_{-d} = 0$$
$$\Rightarrow \int q(x) (s^r_p(x) - s^r_q(x)) \delta(y_d - x^T g) dx = 0$$

where the third equality is from Lemma 7. Then we can follow the same proof technique in Theorem 1 and Remark 1 to show $SK_o(q, p) = 0$ only if $p = q$ a.e.

D Deep Kernel

Assume we have a smooth injective mapping $\phi$, we define the following term

$$\xi_{p,r,g,\phi}(x, \cdot) = s^r_p(x) k_{rg}(\phi^g(x), \cdot) + C_{\phi}(x) \nabla_{\phi^g(x)} k_{rg}(\phi^g(x), \cdot)$$ (38)
We denote

\[ p = \text{validity of the corresponding SKSD discrepancy measure.} \]

Therefore, if \( p \) and substitute this into Stein identity. This returns \( y \).

Similar to the proof in Theorem 3, the RHS term above can be re-written as

\[ DSK_0(q, p) = \sum_{r \in \mathcal{O}_r} \int_{\mathbb{R}^{D-1}} p(y)D^2_{r,g,\phi}(q, p)dy, \quad D^2_{r,g,\phi}(q, p) = E_q[h_{p,r,g,\phi}(x, x')], \quad (40) \]

and it is 0 if and only if \( p = q \) a.e.

Deep-SKSD (Eq. (40)) can be viewed as a generalization of SKSD (Eq. (11)). Specifically, SKSD can be recovered using Deep-SKSD with \( \phi \) as the identity mapping.

D.1 Theorem 4

Proof. We follow the proof of Theorem 2 to show

\[ h_{p,r,g,\phi}(x, y) = \langle \xi_{p,r,g,\phi}(x, \cdot), \xi_{p,r,g,\phi}(y, \cdot) \rangle_{H_{r,g}}. \]

By Assumption 5, \( k_{r,g} \) belongs to the Stein class of \( q \). Thus, \( k_{r,g}(\phi^\delta(x), \cdot) \) belongs to the Stein class of \( q \). This can be easily verified by using the definition of Stein class of \( q \), and the facts that the kernel function is bounded and \( q \) vanishes at boundary. Now we follow the proof in proposition 1 to define

\[ F(x) = \begin{bmatrix} r_1 k_{r,g}(\phi^\delta(x), \cdot) \\ r_2 k_{r,g}(\phi^\delta(x), \cdot) \\ \vdots \\ r_D k_{r,g}(\phi^\delta(x), \cdot) \end{bmatrix} \]

and substitute this into Stein identity. This returns

\[ \int q(x)|s^*_r(x)|k_{r,g}(\phi^\delta(x), \cdot) + C_{r,g,\phi}(x)\nabla_{\phi^\delta(x)} k_{r,g}(\phi^\delta(x), \cdot)\] \( dx = 0 \)

\[ \Rightarrow E_q[\xi_{q,r,g,\phi}(x, \cdot)] = 0 \]

\[ \Rightarrow D^2_{r,g,\phi}(q, q) = E_q[h_{q,r,g,\phi}(x, x')] = (E_q[\xi_{q,r,g,\phi}(x, \cdot)], E_q[\xi_{q,r,g,\phi}(x', \cdot)])_{H_{r,g}} = 0. \]

Therefore, if \( p = q \) a.e., then \( DSK_0(q, p) = 0. \)

Now we prove \( DSK_0(q, p) = 0 \) only if \( p = q \) a.e.. It is trivial that \( DSK_0(q, p) = 0 \) if and only if \( D^2_{r,g,\phi}(q, p) = 0. \) In other words,

\[ DSK_0(q, p) = 0 \quad \Rightarrow \quad E_q[\xi_{p,r,g,\phi}(x, \cdot)] = 0. \]

Similar to the proof in Theorem 2, the RHS term above can be re-written as

\[ E_q[\xi_{p,r,g,\phi}(x, \cdot)] = E_q[(s^*_p(x) - s^*_q(x))k_{r,g}(\phi^\delta(x), \cdot)]. \]

We denote \( G \) as an arbitrary rotation with the \( \delta^\text{th} \) entry as the test direction \( g \). We also define \( y = \phi(x) \) and \( u = Gy \) with \( u_d = y^T g \). Thus, by the change of variable formula, we have

\[ q_\phi(y) = q(x)|J|^{-1}, \]

\[ q_{G\phi}(u) = q_\phi(y)|G|^{-1} = q_\phi(y), \]

(41)
We propose a Goodness-of-fit test method based on the U-statistics of maxSKSD (Eq.(13)) given $p$

When $\phi$ is injective, we have

Assume the conditions in Theorem 3 are satisfied, we have the following:

Finally using similar proof techniques in Theorem 1, we have

We can directly use the results in Section 5.5 of [42]. We only need to check the conditions

where $J$ is the Jacobian matrix $\frac{\partial \phi(x)}{\partial x}$ and $| \cdot |$ is the determinant. Thus, we have

\[
\int q(x)[\nabla x \log \frac{p(x)}{q(x)}]^T r k_{rg}(\phi^\theta(x), \cdot)dx
\]

\[
= \int q_\phi(y)[J][\nabla \phi^{-1}(y) \log \frac{p(\phi^{-1}(y))}{q(\phi^{-1}(y))}]^T r k_{rg}(y^T g, \cdot) |J|^{-1}dy
\]

\[
= \int q_G(\phi^{-1}(G^{-1}u)) \log \frac{p(\phi^{-1}(G^{-1}u))}{q(\phi^{-1}(G^{-1}u))} G_{rg}(u, \cdot) |G^{-1}u| du du_d u_d
\]

\[
= \int q_G(\phi^{-1}(G^{-1}u)) \log \frac{p(\phi^{-1}(G^{-1}u))}{q(\phi^{-1}(G^{-1}u))} H(u_d) du_d
\]

\[
= \int q_\phi(y)[\nabla \phi^{-1}(y) \log \frac{p(\phi^{-1}(y))}{q(\phi^{-1}(y))}]^T r \delta(u_d - y^T g) dy.
\]

As $\phi$ is injective, we have $p = q$ a.e.

\[\square\]

E Applications of maxSKSD

E.1 Goodness-of-fit test

We propose a Goodness-of-fit test method based on the U-statistics of maxSKSD (Eq.(13)) given the optimal test direction $g_r$. In the following we analyze the asymptotic behavior of the proposed statistic.

**Theorem 5.** Assume the conditions in Theorem 3 are satisfied, we have the following:

1. If $q \neq p$, then $\hat{SK}_{max}(q, p)$ is asymptotically normal. Particularly,
   \[
   \sqrt{N}(\hat{SK}_{max}(q, p) - SK_{max}(q, p)) \xrightarrow{d} N(0, \sigma_q^2)
   \]
   where $\sigma_q^2 = \text{var}_{\sim q}(\sum_{r \in O_r} \mathbb{E}_{x \sim q} [h_{p,r,g_r}(x, x')])$ and $\sigma_q \neq 0$

2. If $q = p$, we have a degenerated U-statistics with $\sigma_h = 0$ and
   \[
   N \hat{SK}_{max}(q, p) \xrightarrow{d} \sum_{j=1}^{\infty} c_j (Z_j^2 - 1)
   \]
   where $\{Z_j\}$ are i.i.d standard Gaussian variables, and $\{c_j\}$ are the eigenvalues of the kernel $\sum_{r \in O_r} h_{p,r,g_r}(x, x')$ under $q(x)$. In other words, they are the solutions of $c_j \phi_j(x) = \int x r \sum_{r \in O_r} h_{p,r,g_r}(x, x') \phi_j(x') q(x') dx'$.

**Proof.** We can directly use the results in Section 5.5 of [42]. We only need to check the conditions $\sigma_h \neq 0$ when $p \neq q$ and $\sigma_h = 0$ when $p = q$.

When $p = q$, we re-write $\mathbb{E}_{x \sim q}[h_{p,r,g_r}]$ as

\[
\mathbb{E}_{x \sim q}[h_{p,r,g_r}(x, x')] = (\xi_{p,r,g_r}(x, \cdot), \mathbb{E}_{z \sim q}[\xi_{p,r,g_r}(x', \cdot)]) \varphi_{r,g_r}
\]

From the Eq. (37) in theorem 3 we have $\mathbb{E}_{x \sim q}[\xi_{p,r,g}(x', \cdot)] = 0$. Thus, $\mathbb{E}_{x \sim q}[h_{p,r,g_r}] = 0$ for all $r \in O_r$. Thus, we have $\sigma_h = 0$ when $q = p$.
We assume when \( p \neq q, \sigma_h = 0 \). This means \( \mathbb{E}_x \sim q[h_{p,r,g}(x, x')] = c_r \) where \( c_r \) is a constant. Thus,

\[
c_r = \mathbb{E}_{x \sim p}[\mathbb{E}_{x' \sim q}[h_{p,r,g}(x, x')]] \\
\Rightarrow c_r = \mathbb{E}_{x \sim q}[\mathbb{E}_{x \sim p}[h_{p,r,g}(x, x)]]
\]

From the Eq.(37) in Theorem 3 we have \( c_r = 0 \) for all \( r \in O_r \). Thus, \( \mathbb{E}_{x,x' \sim q}[h_{p,r,g}(x, x')] = c_r = 0 \) which contradict \( p \neq q \).

This theorem indicates a well-defined limit distribution for maxSKSD U-statistics. Next, similar to the previous work \cite{17}, we adopt the bootstrap method \cite{2, 24}. We draw random weights from multinomial distributions, \( (w_1^m, \ldots, w_N^m)_{m=1}^M \sim \text{Multi}(N, \frac{1}{N}, \ldots, \frac{1}{N}) \). The bootstrap sample can be computed

\[
\tilde{SK}_m = \sum_{1 \leq i \neq j \leq N} (w_i^m - \frac{1}{N})(w_j^m - \frac{1}{N}) \sum_{r \in O_r} h_{p,r,g}(x_i, x_j) \quad (44)
\]

The quantile computed by the bootstrap samples is consistent to the one using degenerated U-statistics. This consistence is established in \cite{23, 24}. Algorithm 1 provides a summary of the GOF test method.

**Algorithm 1: GOF Test with maxSKSD U-statistics**

**Input**: Samples \( \{x_i\}_{i=1}^N \sim q(x) \), score function \( s_p(x) \), Orthogonal basis \( O_r \), optimal test direction \( g_r \) for each \( r \in O_r \), kernel function \( k_{rg} \), significant level \( \alpha \), and bootstrap sample size \( M \).

**Hypothesis**: \( H_0: p = q \) v.s. \( H_1: q \neq p \)

Compute \( \tilde{SK}_{max}(q, p) \) using U-statistic Eq.(13);

Generate \( M \) bootstrap samples \( \{\tilde{SK}_m\}_{m=1}^M \) using Eq.(44);

Reject null hypothesis \( H_0 \) if the proportion \( \tilde{SK}_m > \tilde{SK}_{max}(q, p) \) is less than \( \alpha \).

### E.2 Sliced SVGD

The “slicing” idea can also be applied to improve SVGD (Section 2.2). To see this, we modify the flow mapping to \( T_G(x) : \mathbb{R}^D \to \mathbb{R}^D \) as \( T_G(x) = x + \epsilon \phi_G(x) \). Specifically for \( \phi_G(x) \), we adopt \( D \) univariate perturbations instead of one multivariate perturbation:

\[
\phi_G(x) = \begin{bmatrix}
\phi_{g_1}(x^T g_1) \\
\vdots \\
\phi_{g_D}(x^T g_D)
\end{bmatrix} \quad (45)
\]

where \( G = [g_1, \ldots, g_D] \in \mathbb{R}^{D \times D} \) represents slice matrix. For this specific mapping we have the following result analogous to Lemma 1.

**Lemma 8.** Let \( T_G(x) = x + \epsilon \phi_G(x) \) where \( \phi_G \) is defined as Eq.(45). Define \( q_{[T_G]}(z) \) as the density of \( z = T_G(x) \) when \( x \sim q(x) \), with slice matrix \( G \), we have

\[
\nabla_x |q_{[T_G]}(z)|_{\epsilon=0} = - \sum_{d=1}^D \mathbb{E}_q[s^d_p(x)\phi_{g_d}(x^T g_d) + g_{d,d}\nabla_{x^T g_d}\phi_{g_d}(x^T g_d)] 
\]

where \( s^d_p(x) = \nabla_x \log p(x) \) and \( g_{d,d} \) is the \( d \)th element in \( g_d \).

**Proof.** This can be easily verified by substituting Eq.(45) into Eq.(4).

Eq.(46) is similar to maxSSD (Eq.(7)) where the optimal test directions and test functions are replaced with matrix \( G \) and perturbation \( \phi_{g_d}(x) \). \( O_r \) takes the values one-hot vectors. The main difference between this decrease magnitude and maxSSD is that we do not assume \( G \) is optimal. Next, we show how to obtain an analytic descent directions that maximize the decrease magnitude.

By restricting each perturbation \( \phi_{g_d} \in H_{t^d, g_d} \) where \( H_{t^d, g_d} \) is an RKHS equipped with kernel, we have the following result.
Lemma 9. Assume the conditions in lemma[8] If for each perturbation \( \phi_{d} \in \mathcal{H}_{rg_{d}} \) where \( \mathcal{H}_{rg_{d}} \) is an RKHS equipped with kernel \( k_{rg_{d}} \) and \( ||\phi_{d}||_{\mathcal{H}_{rg_{d}}} \leq D_{rg_{d}}(q,p) \), then the steepest descent direction for \( d^{th} \) perturbation is

\[
\phi_{d}^{*}(\cdot) = E_{q}[\xi_{p,r_{d},d_{d}}(x,\cdot)],
\]

and

\[
\nabla, K L[q_{(T_{c})}||p]_{\epsilon=0} = - \sum_{d=1}^{D} D_{d_{d}}^{2}(q,p), \tag{48}
\]

where \( D_{d_{d}}^{2}(q,p) = E_{q}[h_{p,r_{d},d_{d}}(x,x')] \) with one-hot vector \( r_{d} \).

Proof. We show this result using the reproducing property of RKHS \( \mathcal{H}_{rg_{d}} \). The supremum of Eq.(46) can be re-written as

\[
\sup_{\phi_{d}} \sum_{d=1}^{D} E_{q}[s_{p}^{d}(x)\phi_{d}(x^{T}g_{d}) + r_{d}^{T}g_{d}\nabla x^{T}g_{d}\phi_{d}(x^{T}g_{d})]
\]

\[
= \sum_{d=1}^{D} \sup_{\phi_{d}} E_{q}[s_{p}^{d}(x)\phi_{d}(x^{T}g_{d}) + r_{d}^{T}g_{d}\nabla x^{T}g_{d}\phi_{d}(x^{T}g_{d})]
\]

\[
= \sum_{d=1}^{D} \sup_{\phi_{d} \in \mathcal{H}_{rg_{d}}} \frac{E_{q}[s_{p}^{d}(x)\phi_{d}(x^{T}g_{d}) + r_{d}^{T}g_{d}\nabla x^{T}g_{d}\phi_{d}(x^{T}g_{d}), \phi_{d}]_{\mathcal{H}_{rg_{d}}}}{||\phi_{d}||_{\mathcal{H}_{rg_{d}}} \leq D_{rg_{d}}(q,p) \} \tag{49}
\]

\[
= \sum_{d=1}^{D} \sum_{\phi_{d} \in \mathcal{H}_{rg_{d}}} \frac{E_{q}[h_{p,r_{d},d_{d}}(x,x')] \phi_{d}]_{\mathcal{H}_{rg_{d}}}}{||\phi_{d}||_{\mathcal{H}_{rg_{d}}} \leq D_{rg_{d}}(q,p) \} \}
\]

\[
= \sum_{d=1}^{D} E_{q}[h_{p,r_{d},d_{d}}(x,x')] = \sum_{d=1}^{D} D_{d_{d}}^{2}(q,p), \tag{48}
\]

where the third equality is because of the Bochner integrability of \( \xi_{p,r_{d},d_{d}}(x,\cdot) \) shown in Theorem[2].

And the optimal perturbation for \( d^{th} \) dimension is

\[
\phi_{d}^{*}(\cdot) = E_{q}[\xi_{p,r_{d},d_{d}}(x,\cdot)]. \tag{50}
\]

Note that in Lemmas[8] and [9] we assume an arbitrary projection matrix \( G \). To find the steepest descent direction, one can maximize Eq.(48) w.r.t. \( G \). In this case this decrease magnitude Eq.(48) is identical to maxSKSD Eq.(12) with orthogonal basis \( O_{r} \) and optimal test directions \( G \).

The name sliced SVGD comes from that for each perturbation \( \phi_{d_{d}}(\cdot) \), the kernel \( k_{rg_{d}} \) and the repulsive force \( r_{d}^{T}g_{d}\nabla x^{T}g_{d}k_{rg_{d}}(x^{T}g_{d},\cdot) \) are evaluated on \( x^{T}g_{d} \) instead of \( x \) in SVGD. Although S-SVGD only uses one-dimensional projection of \( x \), it is still a valid inference method as long as the optimality of \( G \) is ensured, because maxSKSD is a valid discrepancy measure[2]. The S-SVGD method is summarised in Algorithm[2]. In practice this algorithm may violate the optimality condition of \( G \) (due to estimation error using finite samples and local optimum found by

\[\text{Note that maximizing Eq.(48) w.r.t. sliced matrix } G \text{ is necessary, otherwise Eq.(48) is not a valid discrepancy measure, and a zero value does not imply } p = q. \text{ In such case the resulting particle inference method is not asymptotically exact.} \]
gradient-based optimization), which is a common issue in many adversarial training procedure.

Algorithm 2: S-SVGD for variational inference

Input : Initial samples \( \{ x_i \}_{i=1}^{N} \), target score function \( s_p(x) \), Orthogonal basis \( O_r \), initial slice matrix \( G \), kernel function \( k_{rg} \), iteration number \( L \) and step size \( \epsilon \).

Output : Set of particles \( \{ x_i \}_{i=1}^{N} \) that approximates \( p \)

for \( l \leq L \) do
    Update each particles \( x_i^{l+1} = x_i^l + \epsilon \phi_{G}^*(x_i^l) \) where \( \phi_{G}^*(x_i^l) \) is computed using Eq. (47);
    Find the optimal slice matrix \( G \) by maximizing Eq. (48) using \( \{ x_i^{l+1} \}_{i=1}^{N} \);
end

F GOF test

F.1 Setup for High dimensional benchmark GOF test

For each GOF test, we draw 1000 samples from alternative hypothesis \( q \). These samples are directly used for GOF test methods that do not require any training procedure, like KSD, MMD, RFSD and FSSD Rand. However, for methods that require training like maxSKSD and FSSD Opt, we split the entire samples into 200 training data and 800 GOF test data as \([27, 23]\). For maxSKSD, we initially draw the slice matrix \( G \) from a normal distribution before normalizing the magnitude of each vectors in \( G \) to 1 and use Adam with learning rate 0.001 to update it (maximizing Eq. (12)). For FSSD-Opt, we use the default settings in the original publication. During the GOF test, only the test data are used for FSSD-Opt and maxSKSD. We set the significant level \( \alpha = 0.05 \) and the dimension of the distribution grows from 2 to 100. We use 1000 bootstrap samples for all tests, and 1000 trials for Gaussian Null test, 500 trials for Gaussian Laplace test, 250 trials for Gaussian Multivariatet-t test and 500 trials for Gaussian diffusion test.

F.2 Setup for RBM GOF test

For the RBM, we use 50 dimension for observable variable and 40 dimension for hidden variable. We run 100 trials with 1000 bootstrap samples for each method. 1000 test samples are used for methods like KSD, MMD, RFSD and FSSD Rand. For maxSKSD_g, maxSKSD_rg and FFSD Opt that require training, we use 800 samples for test. Parallel block Gibbs sampler with 2000 burn-in is used to draw samples from \( q \). To avoid the over-fitting to small training samples, we use 200 samples to update the slice matrix \( G \) (or \( G \) and \( r \) for maxSKSD_rg) in each Gibbs step during the burning. However, it should be noted that these intermediate samples from the burn-in should not be used as the test samples for other methods because they are not from \( q \). This setup is slightly different from the most general GOF test where only test samples and target density are given. However, it is still useful for some applications such as detecting convergence/selecting hyper-parameter of MCMC sampler (appendix F.3). Finding relatively good directions for maxSKSD with fewer training samples is a good direction for future work.

F.3 Selecting hyperparameter of a biased sampler

We use the proposed methods to select the step size of a biased sampler. Particularly, we consider using SGHMC here which is a biased sampler without Metropolis-Hasting step. The bias is mainly caused by the discretization error, namely, the step size. For smaller step size, the bias is small but the mixing speed is slow. Larger step size results in higher bias with fast mixing.

Selecting the step size is essentially a GOF test problem, where alternative hypothesis is the invariant underlying distribution of SGHMC, and the bias is quantified by the discrepancy value. The best step size is the one corresponding to the lowest discrepancy value. We compare our proposed maxSKSD based methods with KSD. The target distribution is a 15 dimensional correlated Gaussian distribution with zero mean and randomly generated co-variance matrix. We also include a strong baseline using KL divergence where the \( q \) is a Gaussian distribution, with parameters estimated by samples from SGHMC.
Figure 4: Discrepancy value for different random seed. The x-axis indicates the step size used for SGHMC.

**Setup** We run 100 parallel SGHMC chains with 2000 burn-in period. During each step in burn-in, we update the sliced matrix $G$ (and $r$) using such 100 samples. After burn-in, we fix the $G$ and $r$ and continue to run SGHMC with thinning 5 until 1500 test samples are collected. We run this experiment using 3 different seeds. For KSD and maxSKSD discrepancy value, we use U-statistics due to its unbiasedness.

Figure 4 shows the discrepancy curve with different step sizes and Table 4 shows some diagnostic statistics. KL based method is used as the ‘ground truth’ measure. In summary, the step sizes chosen by maxSKSD based methods are more sensible than those selected by KSD. To be specific, take random seed 1 as an example, KSD failed to detect the non-convergence for step size larger than 0.011 where KL starts to increase. Even worse, KSD achieves the lowest value at step size 0.015 which is a poor choice indicated by KL divergence. On the other hand, maxSKSD based methods, especially maxSKSD_rg, can detect the non-convergence and agrees with the trend shown by the KL method. The above also holds true for other random seeds.

| Method   | Random seed |
|----------|-------------|
|          | 1   | 2   | 3   |
| KL       | 0.004| 0.004| 0.004|
| KSD      | 0.015| 0.015| 0.013|
| maxSKSD_rg | 0.004| 0.008| 0.008|
| maxSKSD_g | 0.013| 0.004| 0.008|

Table 4: **Top:** This table shows the step size chosen by different methods. We can observe that those chosen by maxSKSD based methods and KL method are closer compared to KSD. **Bottom:** The divergence value at the chosen step size for random seed 2. The row indicates the method used to choose the step size and column indicates the corresponding values. We can observe maxSKSD based methods indeed agree more with KL method where the KL value is around 0.2 at step size chosen by maxSKSD based methods. On the other hand, KSD failed to detect the non-convergence at step size 0.015 where KL value is already 1.46.
We increase the dimensions for ICA from 10 to 200 to evaluate their performance in low and high dimensions. We generate the training and test data by using a randomly sampled weight matrix. We use 20000 training data and 5000 test data. To make the computation stabler, we follow [18] such that the weight matrix is initialized until its conditional number is smaller than the dimension of the matrix. For LSD, we follow the exact same architecture as the original paper [18]. For KSD, we use the U-statistics with the bandwidth chosen as the median distance (the training for KSD with V-statistic diverges). For maxSKSD, we instead use the V-statistics with the bandwidth chosen as the median distance (for detailed analysis of this behavior, refer to [51]). To verify this, we plot the Particle Average Repulsive Force (PARF) and the averaged estimated mean of the samples in figure 5. The PARF for SVGD reduces as the dimension increases whereas S-SVGD stays at a constant level. This is because the weight matrix is initialized until its conditional number is smaller than the dimension of the matrix. For maxSKSD, we instead use the V-statistics with 1.5 times median distance as the bandwidth. We train the ICA model for 15000 steps using Adam optimizer with 0.001 learning rate and $\beta_1 = 0.5$, $\beta_2 = 0.9$. We use 5 independent runs and average their results.

G.3 ICA Additional Plots

From the figure [6] we observe at low dimensions ($D = 10$), LSD converges fastest and KSD is the slowest. However, as the dimension increases, the convergence speed of maxSKSD catches up.
Figure 6: Training curve of different methods for ICA problems. The y-axis indicates the NLL of test data.

with LSD and becomes faster after \( D = 60 \), whereas KSD starts to slow down and even diverges at \( D = 200 \).

G.4 Amortized SVGD

Algorithm 3 shows the training framework of amortized SVGD. For experiment details, we use fully connected neural network with ReLU activations and 2 hidden layers for encoder and decoder ([300, 200] and [200, 300] respectively). For decoder output, we use sigmoid activation function and binary cross-entropy for the decoder loss. For the implicit encoder, the input is simply a concatenation of the image and Gaussian noise with the same dimension as the latent space. We also use dropout with probability 0.3 for each layer of the encoder. For SVGD and S-SVGD, we use 0.1 for step size and only run 1 update of the latent samples before we update the encoder. The kernel bandwidth is chosen by the median heuristic. We update the sliced matrix \( G \) for S-SVGD once per each encoder update. 50 latent samples are used for both encoder and decoder updates. We use Adam optimizer\[29\] with 0.001 learning rate and 100 for batch size.

For evaluation, the log likelihood is computed using Hamiltonian annealed importance sampling (HAIS)\[50\]. Specifically, we use 1000 annealed steps and 10 leapfrog update per step. We tune the HAIS step size to maintain 0.65 acceptance rate.

For imputation, we follow \[41\] to use approximate Gibbs sampler with \( D = 32 \) latent space. Specifically, with missing and observed pixels denoted as \( x_m \) and \( x_o \), encoder distribution \( q_\phi \) and decoder \( p_\theta \), we iteratively applies the following procedure: (1) generate latent samples \( z \sim q_\phi(z|x_o, x_m) \) (2) reconstruction \( x^* \sim p_\theta(x^*|z) \) (3) Imputation \( x_m \leftarrow x^*_m \). To compute label entropy and accuracy, 200 parallel samplers are used for each image with 500 steps to make sure they fully converged. The imputation label is found by the nearest neighbour method in training data.
Figure 7: Imputation images after 500 Gibbs steps. Those images are generated by parallel pseudo-
Gibbs sampler. The first column shows the original images. The second column represents the
masked images. The rest of the columns are the imputed images.

Label entropy is computed by its empirical probability and the accuracy is the percentage of the
correct ones among all imputed images.

Algorithm 3: Amortized SVGD

Input: Total training step $T$, Adam learning rate $\epsilon_O$, SVGD/S-SVGD step size $\epsilon_S$, latent sample
size $N$, encoder network $f_q$, decoder network $f_d$ and decoder loss $L$

for $t \leq T$ do
    Generate $N$ initial latent samples using encoder $\{z_i\}_{i=1}^{N} = f_q(x)$;
    Update the samples $\{z^*_i\}_{i=1}^{N}$ based on $\{z_i\}_{i=1}^{N}$ using SVGD or S-SVGD (algorithm 2) with step
    size $\epsilon_S$;
    Compute the encoder MSE loss between $\{z^*_i\}_{i=1}^{N}$ and $\{z_i\}_{i=1}^{N}$ and update encoder $f_q$ using
    Adam($f_q, \epsilon_O$);
    Compute decoder loss $L(x, \{z^*_i\}_{i=1}^{N})$ and update decoder using Adam($f_d, \epsilon_O$);
end

Figure 7 shows some of the resulting imputed images after 500 Gibbs steps. We can clearly observe
that the S-SVGD generated more diverse images compared to Vanilla VAE (e.g. digit ‘8’ and digit
‘5’), where it only captures a single mode. Compared to amortized SVGD, the diversity of generated
images are similar, but the imputed images of S-SVGD seems to be closer to the original image
(e.g. digit ‘8’ and the first digit ‘5’). This explains the high accuracy value in table 3. Although
vanilla VAE also generates images that are close to the original one, it may fail to capture the correct
mode and get stuck at the wrong one (e.g. first digit ‘5’). This explains the slightly worse accuracy
compared to amortized S-SVGD.

G.5 Bayesian Neural Network Regression

We also compare our proposed S-SVGD algorithm with the baseline SVGD in high dimensional
Bayesian neural network inference. We follow the same settings in [36] to use a fully connected
one-hidden-layer neural network with ReLU activation and 50 hidden units. The dataset are randomly
split into 90% training and 10% test data. Batch size 100 is used for all data sets. Each results are
averaged over 15 random trials, except for Protein where 5 trails are conducted. AdaGrad is used
Table 5: BNN results on UCI regression benchmarks, comparing SVGD and S-SVGD. See main text for details.

| Dataset  | SVGD       | RMSE       | SVGD       | test LL | Dist |
|----------|------------|------------|------------|---------|------|
| Boston   | 2.937 ± 0.173 | 2.87 ± 0.163 | -2.534 ± 0.092 | -2.507 ± 0.086 | 54372 ± 986 |
| Concrete | 5.189 ± 0.115 | 4.880 ± 0.082 | -3.076 ± 0.024 | -3.004 ± 0.023 | 54372 ± 986 |
| Combined | 3.979 ± 0.040 | 3.914 ± 0.041 | -2.892 ± 0.010 | -2.786 ± 0.010 | 54372 ± 986 |
| Naval    | 0.0030 ± 0   | 0.0029 ± 0  | 4.368 ± 0.014 | 4.411 ± 0.010 | 54372 ± 986 |
| Wine     | 0.607 ± 0.009 | 0.603 ± 0.009 | -0.924 ± 0.015 | -0.914 ± 0.015 | 54372 ± 986 |
| Energy   | 1.353 ± 0.049 | 1.132 ± 0.048 | -1.736 ± 0.040 | -1.540 ± 0.044 | 54372 ± 986 |
| kin8nm   | 0.082 ± 0.001 | 0.079 ± 0.00 | 1.084 ± 0.012 | 1.104 ± 0.006 | 54372 ± 986 |
| Yacht    | 0.714 ± 0.078 | 0.613 ± 0.064 | -1.277 ± 0.155 | -0.999 ± 0.087 | 54372 ± 986 |
| Protein  | 4.543 ± 0.010 | 4.587 ± 0.009 | -2.932 ± 0.003 | -2.942 ± 0.002 | 54372 ± 986 |

for both SVGD and S-SVGD. For SVGD, the bandwidth is selected in the same way as [36]. For S-SVGD, we use the same way to select the bandwidth except we multiply a coefficient 0.15 in front of the bandwidth. 50 samples are used for both SVGD and S-SVGD. We initialize the particles to be closed to each other. For small datasets like Boston Housing, Yacht and Energy, we apply a small coefficient for the initial repulsive force of S-SVGD, and it gradually increases to 1 after 500, 1000 and 500 epochs respectively. This is to avoid the over-dominance of the repulsive force at the beginning. For other datasets, we do not tune the repulsive force. For Boston Housing, Concrete, and Energy, we train the network for 2000 epochs. We use 500 and 50 epochs for Wine and Protein respectively. For the rest of the data set, we use 200 epochs.

We evaluate the performance through the log likelihood and root mean squared error (RMSE) of the test set, together with the particle-sum distance \( \sum_{1 \leq i < j \leq N} dist(x_i, x_j) \) to examine the spread of the resulting particles. Table 5 shows the performance of BNN trained using SVGD and S-SVGD on 9 UCI data sets. We can clearly observe S-SVGD outperforms SVGD on 7 out of 9 data sets. From the particle-sum distance, the resulting particles from S-SVGD are more spread out than SVGD to prevent mode collapse. This behavior can indeed bring benefits especially when dealing with small data set where uncertainty quantification is important. To be specific, SVGD achieves better result only on the large Protein data set where the epistemic uncertainty is low compared to small data set. Therefore, the mode collapse of SVGD does not affect the performance too much. This can be partially verified by examining other smaller datasets. Boston Housing, Concrete, Energy and Yacht are very small data sets with quite noisy features. Thus, S-SVGD significantly outperforms SVGD on those data sets due to its better uncertainty estimation. For the remaining data set, e.g. Combined, Naval and kin8nm, their data set sizes are between the aforementioned small set and Protein. Thus, S-SVGD still achieves better results but the difference is less significant. One exception is Wine, a small data set, where S-SVGD has similar performance as SVGD. This is because Wine has relatively easy prediction targets.