

## A DEFINITIONS AND ASSUMPTIONS

**Definition A.1. (Stein Class (Liu et al., 2016))** Assume distribution  $q$  has continuous and differentiable density  $q(\mathbf{x})$ . A function  $f$  defined on the domain  $\mathcal{X} \subset \mathbb{R}^D$ ,  $f : \mathcal{X} \rightarrow \mathbb{R}$  is in the **Stein class** of  $q$  if  $f$  is smooth and satisfies

$$\int_{\mathcal{X}} \nabla_x (f(\mathbf{x})q(\mathbf{x}))d\mathbf{x} = 0 \quad (14)$$

We can easily see that the above holds true for  $\mathcal{X} = \mathbb{R}^D$  if

$$\lim_{\|\mathbf{x}\| \rightarrow \infty} q(\mathbf{x})f(\mathbf{x}) = 0 \quad (15)$$

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

**Definition A.2. (Liu et al., 2016)** A kernel  $k(\mathbf{x}, \mathbf{x}')$  is said to be in the Stein class of  $q$  if  $k(\mathbf{x}, \mathbf{x}')$  has continuous second order partial derivatives, and both  $k(\mathbf{x}, \cdot)$  and  $k(\cdot, \mathbf{x})$  are in the Stein class of  $q$  for any fixed  $\mathbf{x}$

**Radon Transform** In machine learning literature, Radon transform has been used as the primary tool to derive sliced Wasserstein distance (Kolouri et al., 2019; Deshpande et al., 2019; 2018). To be specific, the standard Radon transform, denoted as  $\mathcal{R}$ , 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:

$$L^1(\mathbb{R}^D) = \{I : \mathbb{R}^D \rightarrow \mathbb{R} \mid \int_{\mathbb{R}^D} |I(\mathbf{x})|d\mathbf{x} < \infty\}, \quad (16)$$

the Radon transform is defined by

$$\mathcal{R}[I](l, \mathbf{g}) = \int_{\mathbb{R}^D} I(\mathbf{x})\delta(l - \langle \mathbf{x}, \mathbf{g} \rangle)d\mathbf{x} \quad (17)$$

for  $(l, \mathbf{g}) \in \mathbb{R} \times \mathbb{S}^{D-1}$  where  $\mathbb{S}^{D-1} \subset \mathbb{R}^D$  stands for a unit sphere in  $\mathbb{R}^D$ . For fixed  $\mathbf{g}$ , this defines a continuous function  $\mathcal{R}[I](\cdot, \mathbf{g}) : \mathbb{R} \rightarrow \mathbb{R}$  which is the projection of function  $I$  on to the hyper-plane with its normal vector defined by  $\mathbf{g}$  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  $p(\mathbf{x}), q(\mathbf{x})$  supported on  $\mathbb{R}^D$ . Density  $q$  satisfies:  $\lim_{\|\mathbf{x}\| \rightarrow \infty} q(\mathbf{x}) = 0$ .

**Assumption 2** (Regularity of score functions) Denote the score function of  $p(\mathbf{x})$  as  $\mathbf{s}_p(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) \in \mathbb{R}^D$  and score function of  $q(\mathbf{x})$  accordingly. Assume the score functions satisfy

$$\begin{aligned} \int_{\mathbb{R}^D} q(\mathbf{x})|(\mathbf{s}_p(\mathbf{x}) - \mathbf{s}_q(\mathbf{x}))^T \mathbf{r}|d\mathbf{x} &< \infty \\ \int_{\mathbb{R}^D} q(\mathbf{x})\|(\mathbf{s}_p(\mathbf{x}) - \mathbf{s}_q(\mathbf{x}))^T \mathbf{r}\|^2 d\mathbf{x} &< \infty \end{aligned} \quad (18)$$

for all  $\mathbf{r}$  where  $\mathbf{r} \in \mathbb{S}^{D-1}$  is a vector sampled from a uniform distribution over a unit ball  $\mathbb{S}^{D-1}$ . In other words, the score difference, when projected on the  $\mathbf{r}$  direction, is both  $L^1$  and  $L^2$  integrable with respect to the probability measure defined by  $q(\mathbf{x})d\mathbf{x}$ . 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  $f(\cdot; \mathbf{r}, \mathbf{g}) : \mathbb{R}^D \rightarrow \mathbb{R}$  is smooth and belongs to the Stein class of  $q$ .

**Assumption 4** (Bounded Radon transformed functions) Define

$$I_{q,p} = q(\mathbf{x})(s_p(\mathbf{x}) - s_q(\mathbf{x}))^T \mathbf{r} \quad (19)$$

We assume the Radon transformation of  $I_{q,p}$ ,  $\mathcal{R}[I_{q,p}](l, \mathbf{g})$  is bounded for all  $\mathbf{g}$ , where  $\mathbf{g}$  is sampled from a uniform distribution over a unit ball  $\mathbb{S}^{D-1}$ . Namely,  $\|\mathcal{R}[I_{q,p}](l, \mathbf{g})\|_\infty < \infty$

**Assumption 5** (Properties of kernels) For the RKHS  $\mathcal{H}_{r,g}$  equipped with kernel function  $k(\cdot, \cdot; \mathbf{r}, \mathbf{g})$  defined as  $k(\mathbf{x}, \mathbf{x}'; \mathbf{r}, \mathbf{g}) = k_{rg}(\mathbf{x}^T \mathbf{g}, \mathbf{x}'^T \mathbf{g})$ , we assume the kernel  $k_{rg}$  is  $C_0$ -universal and  $k$  belongs to the Stein class of  $q$  for all  $\mathbf{r}$  and  $\mathbf{g}$ . We further assume the kernel  $k_{rg}$  is uniformly bounded such that  $\sup_{\mathbf{x}} k_{rg}(\mathbf{x}^T \mathbf{g}, \mathbf{x}^T \mathbf{g}) < \infty$  for all  $\mathbf{g}$  and  $\mathbf{r}$ . For example, the RBF kernel is a suitable choice for  $k_{rg}$ .

## B SSD RELATED PROPOSITIONS AND THEOREMS

### B.1 UNDERSTANDING THE PROPOSED SLICE STEIN DISCREPANCIES

We provide an explanation on the roles played by  $\mathbf{r}$  and  $\mathbf{g}$ . We follow the same notations used in defining SSD (Eq.(5)). The key idea to tackle the curse-of-dimensionality is to project both the score function  $s_p(\mathbf{x}) \in \mathbb{R}^D$  and test function input  $\mathbf{x} \in \mathbb{R}^D$ . First, the *slicing direction*  $\mathbf{r}$  is introduced to project the score function, i.e.  $s_p^r(\mathbf{x}) = s_p(\mathbf{x})^T \mathbf{r}$ . By doing so, if  $s_p^r(\mathbf{x}) = s_q^r(\mathbf{x})$  for all  $\mathbf{r} \in \mathbb{S}^{D-1}$ , then  $s_p(\mathbf{x}) = s_q(\mathbf{x})$  and  $p = q$  a.e. These equality conditions can be checked using Stein discrepancy (section 2.1) by replacing  $s_p(\mathbf{x})$  with  $s_p^r(\mathbf{x})$ . Now it remains to address the scalability issue for the test functions as the score projection operation does not reduce the dimensionality of the test function input  $\mathbf{x}$ . In fact, using similar ideas from section 2.1, the optimal test function to describe the difference between the projected score is proportional to  $s_p^r(\mathbf{x}) - s_q^r(\mathbf{x})$ , which is an  $\mathbb{R}^D \rightarrow \mathbb{R}$  function and thus it still utilizes the information in the original space  $\mathbf{x} \in \mathbb{R}^D$ .

To resolve the high dimensionality of  $\mathbf{x}$ , it is preferred to use a test function that is defined on the one-dimensional input  $\mathbb{R} \rightarrow \mathbb{R}$ . However, using the projected input along the slicing direction  $\mathbf{r}$  for the test function is insufficient to tell differences between the projected scores due to the information loss, as shown in the pathological example at the end of this section. Therefore, we need to find a way to express the projected score difference using a wide range of one-dimensional representations.

Our solution takes inspiration from the idea of CT-scans. To be precise, we test the difference of the projected score along a *test direction*  $\mathbf{g}$ , by projecting  $q(\mathbf{x})(s_p^r(\mathbf{x}) - s_q^r(\mathbf{x}))$  to direction  $\mathbf{g}$ . This is exactly the Radon transform of function  $q(\mathbf{x})(s_p^r(\mathbf{x}) - s_q^r(\mathbf{x}))$  in the direction  $\mathbf{g}$ , which is an  $\mathbb{R} \rightarrow \mathbb{R}$  function with input  $\mathbf{x}^T \mathbf{g}$ . Importantly, if the Radon transformed projected score difference is zero for all  $\mathbf{g}$ , the invertibility of Radon transform tell us the projected score difference is zero, and then  $p = q$  a.e. if it holds true for all  $\mathbf{r}$ . Again this equality condition for the Radon transformed projected score difference can be checked in a similar way as in Stein discrepancy by defining test functions with input  $\mathbf{x}^T \mathbf{g}$ .

To see why using a test direction  $\mathbf{g}$  is necessary, we provide a counter-example in the case of using orthonormal slicing basis  $O_r$ . That is, if we set  $\mathbf{g} = \mathbf{r}$  for  $\mathbf{r} \in O_r$ , then there exists a pair of distributions  $p \neq q$  such that the following discrepancy equals to zero:

$$D(q, p) = \sum_{\mathbf{r} \in O_r} \sup_{f_r \in \mathcal{F}_q} \mathbb{E}_q[s_p^r(\mathbf{x}) f_r(\mathbf{x}^T \mathbf{r}) + \nabla_{\mathbf{x}^T \mathbf{r}} f_r(\mathbf{x}^T \mathbf{r})]. \quad (20)$$

To see this, we first select  $O_r$  to be the standard orthonormal basis of  $\mathbb{R}^D$  (i.e. the basis formed by one-hot vectors) w.l.o.g., as all the orthonormal basis in  $\mathbb{R}^D$  are equivalent up to rotations or reflections. Now consider two probability distributions  $p$  and  $q$  supported on  $\mathbb{R}^D$ , where  $p(\mathbf{x}) = \prod_i^D p(x_i)$  and  $q(x_i) = p(x_i)$ . Importantly,  $q$  distribution might not be factorized. Then we have

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

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 1

We split the proof of theorem 1 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 (Stein et al., 2004; Liu et al., 2016).

**Lemma 2.** (*Stein Identity*) *Assume  $q$  is a smooth density satisfied assumption 1, then we have*

$$\mathbb{E}_q[\mathbf{s}_q(\mathbf{x})f(\mathbf{x})^T + \nabla f(\mathbf{x})] = \mathbf{0}$$

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

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

$$\int q(\mathbf{x})[\mathbf{s}_q(\mathbf{x})^T F(\mathbf{x}) + \nabla_{\mathbf{x}}^T F(\mathbf{x})] d\mathbf{x} = 0$$

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

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

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

where  $\mathbf{r} = [r_1, r_2, \dots, r_D]^T$ .

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

$$\begin{aligned} \int q(\mathbf{x})[\mathbf{s}_q(\mathbf{x})^T F(\mathbf{x}) + \nabla_{\mathbf{x}}^T F(\mathbf{x})] d\mathbf{x} &= 0 \\ \Rightarrow \int q(\mathbf{x})[\mathbf{s}_q(\mathbf{x})^T \mathbf{r} f_{\mathbf{r},\mathbf{g}}(\mathbf{x}^T \mathbf{g}) + \mathbf{r}^T \mathbf{g} \nabla_{\mathbf{x}^T \mathbf{g}} f_{\mathbf{r},\mathbf{g}}(\mathbf{x}^T \mathbf{g})] d\mathbf{x} &= 0 \end{aligned}$$

for all test functions  $f_{\mathbf{r},\mathbf{g}}$  that belongs to Stein class of  $q$ . Therefore, Eq.(5) is 0 if  $p = q$  a.e.  $\square$

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 3.** (*Fourier Slice Theorem (Bracewell, 1956)*) *For a particular smooth function  $f(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}$  that satisfies assumptions of Radon transforms, we define  $\mathcal{F}_D$  as the  $D$  dimensional Fourier transform operator,  $\mathcal{S}_1$  as a slice operator which extracts 1 dimensional central slice of a function and  $\mathcal{R}$  as the Radon transform operator. Thus, for a slice direction  $\mathbf{g}$ , we have the following equivalence*

$$\mathcal{S}_1[\mathcal{F}_D[f]](\omega, \mathbf{g}) = \mathcal{F}_1[\mathcal{R}[f](l, \mathbf{g})](\omega). \quad (22)$$

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  $\mathbf{g}$  from the transformed function.
- First apply the Radon transform with direction  $\mathbf{g}$  to the function  $f$  and then apply one dimensional Fourier transform to the projected function.

Next we show some properties related to the rotated or reflected distributions.

**Lemma 4.** (*Marginalization Invariance of rotated distribution*) Assume we have a probability distribution  $q$  supported on  $\mathbb{R}^D$ , a orthogonal matrix  $\mathbf{G} \in \mathbb{R}^{D \times D}$  and a test function  $f : \mathbb{R}^D \rightarrow \mathbb{R}$ , we can define the corresponding rotated distribution  $q_G$  after applying the rotation matrix  $\mathbf{G}$ . Thus, we have the following identity

$$\int q_G(\mathbf{x}) f(\mathbf{G}^{-1}\mathbf{x}) d\mathbf{x} = \int q(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}. \quad (23)$$

*Proof.* By the definition of rotation and change of variable formula, we define  $\mathbf{y} = \mathbf{G}\mathbf{x}$ , we can show

$$\begin{aligned} q_G(\mathbf{y}) &= q(\mathbf{x}) |\mathbf{G}^{-1}| \\ &= q(\mathbf{G}^{-1}\mathbf{y}) \times 1 \\ &= q(\mathbf{G}^{-1}\mathbf{y}) \end{aligned}$$

where  $|\mathbf{G}^{-1}|$  represents the determinant of the inverse rotation matrix. Thus, by change of variable formula, we have

$$\begin{aligned} &\int q_G(\mathbf{y}) f(\mathbf{G}^{-1}\mathbf{y}) d\mathbf{y} \\ &= \int q(\mathbf{G}^{-1}\mathbf{y}) f(\mathbf{G}^{-1}\mathbf{y}) d\mathbf{y} \\ &= \int q(\mathbf{x}) f(\mathbf{x}) |\mathbf{G}| d\mathbf{x} \\ &= \int q(\mathbf{y}) f(\mathbf{y}) d\mathbf{y}. \end{aligned}$$

□

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 5.** (*Change of Variable Formula using Matrix Volume (Ben-Israel, 1999)*) If  $\mathcal{U}$  and  $\mathcal{V}$  are sets in spaces with different dimensions, say  $\mathcal{U} \in \mathbb{R}^n$  and  $\mathcal{V} \in \mathbb{R}^m$  with  $n > m$ , and  $\phi : \mathcal{U} \rightarrow \mathcal{V}$  is a continuously differentiable injective function and  $f : \mathbb{R}^m \rightarrow \mathbb{R}$  is integrable on  $\mathcal{V}$ , we have the following change of variable formula:

$$\int_{\mathcal{V}} f(\mathbf{v}) d\mathbf{v} = \int_{\mathcal{U}} (f \circ \phi)(\mathbf{u}) \text{vol} J_{\phi}(\mathbf{u}) d\mathbf{u} \quad (24)$$

where  $\text{vol} J_{\phi}(\mathbf{u})$  is the matrix volume of the Jacobian matrix  $J_{\phi}(\mathbf{u}) = \partial(v_1, \dots, v_m) / \partial(u_1, \dots, u_n)$ . Particularly, if  $J_{\phi}(\mathbf{u})$  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 6.** (*Conditional Expectation = Radon Transform*) For a particular test direction  $\mathbf{g}_d \in \mathbb{S}^{D-1}$ , we can define an arbitrary rotation matrix  $\mathbf{G} \in \mathbb{R}^{D \times D}$  that the  $d^{\text{th}}$  entry is the test direction  $\mathbf{g}_d$ . We assume the probability distribution  $q(\mathbf{x})$  is supported on  $\mathbb{R}^D$ , and  $\mathbf{x}_{-d}$  represents  $\mathbf{x} \setminus x_d$  (all elements of  $\mathbf{x}$  except  $x_d$ ). Further, let define the mapping  $\mathbf{x} = \mathbf{G}\mathbf{u}$  for  $\mathbf{u} \in \mathbb{R}^D$  and  $x_d$  is a constant  $x_d = p$ . 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:

$$\int_{\mathcal{X}_d} q_G(x_d, \mathbf{x}_{-d}) f(\mathbf{G}^{-1}\mathbf{x}) d\mathbf{x}_{-d} = \int q(\mathbf{u}) f(\mathbf{u}) \delta(p - \mathbf{u}^T \mathbf{g}_d) d\mathbf{u} \quad (25)$$

where  $\mathcal{X}_d = \{\mathbf{x} \in \mathbb{R}^D | x_d = p\}$ .

*Proof.* From the definition of  $\mathbf{x} = \mathbf{G}\mathbf{u}$ , we can define the rotation matrix  $\mathbf{G}$  as following:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1^T \\ \vdots \\ \mathbf{g}_d^T \\ \vdots \\ \mathbf{g}_D^T \end{bmatrix}$$

where  $\mathbf{g}_d = [g_{d,1}, \dots, g_{d,D}]^T$ . Thus, assume  $x_d = p$ , we can write down

$$\mathbf{x} = \mathbf{G}\mathbf{u} = \begin{bmatrix} \sum_{k=1}^D g_{1,k} u_k \\ \vdots \\ p \\ \vdots \\ \sum_{k=1}^D g_{D,k} u_k \end{bmatrix}.$$

Thus, the Jacobian matrix can be written as

$$J = \frac{\partial(\mathbf{G}\mathbf{u})}{\partial \mathbf{u}} = \begin{bmatrix} g_{1,1} & \dots & g_{d-1,1} & g_{d+1,1} & \dots & g_{D,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{1,D} & \dots & g_{d-1,D} & g_{d+1,D} & \dots & g_{D,D} \end{bmatrix} = [\mathbf{g}_1 \quad \dots \quad \mathbf{g}_{d-1} \quad \mathbf{g}_{d+1} \quad \dots \quad \mathbf{g}_D].$$

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

$$\text{vol} J = \sqrt{\det J^T J} = \begin{bmatrix} \mathbf{g}_1^T \\ \vdots \\ \mathbf{g}_{d-1}^T \\ \mathbf{g}_{d+1}^T \\ \vdots \\ \mathbf{g}_D^T \end{bmatrix} [\mathbf{g}_1 \quad \dots \quad \mathbf{g}_{d-1} \quad \mathbf{g}_{d+1} \quad \dots \quad \mathbf{g}_D] = \mathbf{I},$$

where  $\mathbf{I} \in \mathbb{R}^{(D-1) \times (D-1)}$  is the identity matrix. Then we directly apply the results in Lemma 5 and Lemma 4, we have

$$\begin{aligned} & \int_{\mathcal{X}_d} q_G(x_d, \mathbf{x}_{-d}) f(\mathbf{G}^{-1} \mathbf{x}) d\mathbf{x}_{-d} \\ &= \int_{\mathcal{X}_d} q(\mathbf{G}^{-1} \mathbf{x}) f(\mathbf{G}^{-1} \mathbf{x}) \times 1 d\mathbf{x}_{-d} \\ &\stackrel{\text{Lemma 5}}{=} \int_{\mathcal{U}} q(\mathbf{u}) f(\mathbf{u}) \text{vol} J d\mathbf{u} \end{aligned}$$

where  $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^D | \mathbf{g}_d^T \mathbf{u} = p\}$ . Thus, we have

$$\int_{\mathcal{X}_d} q_G(x_d, \mathbf{x}_{-d}) f(\mathbf{G}^{-1} \mathbf{x}) d\mathbf{x}_{-d} = \int q(\mathbf{u}) f(\mathbf{u}) \delta(p - \mathbf{u}^T \mathbf{g}_d) d\mathbf{u}$$

□

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

*Proof.* In order to prove equation (Eq.(5)) being 0 implies  $p = q$  a.e., the strategy is to construct a lower bound for Eq.(5) 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.(5) is 0, it implies the lower bound is 0 and  $q = p$  a.e.

Consider the inner supreme inside the Eq.(5), by Proposition 1, we have

$$\begin{aligned} & \mathbb{E}_q[\mathbf{s}_p(\mathbf{x})^T \mathbf{r} f_{r,g}(\mathbf{x}^T \mathbf{g}) + \mathbf{r}^T \mathbf{g} \nabla_{\mathbf{x}^T \mathbf{g}} f_{r,g}(\mathbf{x}^T \mathbf{g})] \\ &= \mathbb{E}_q[(\mathbf{s}_p(\mathbf{x}) - \mathbf{s}_q(\mathbf{x}))^T \mathbf{r} f_{r,g}(\mathbf{x}^T \mathbf{g})]. \end{aligned} \tag{26}$$

Now we apply the Lemma 4, and assume  $\mathbf{y} = \mathbf{G}\mathbf{x}$  and  $y_d = \mathbf{g}^T \mathbf{x}$ , then, Eq.(26) can be rewritten as

$$\begin{aligned} & \int q(\mathbf{x})[\mathbf{s}_p(\mathbf{x}) - \mathbf{s}_q(\mathbf{x})]^T \mathbf{r} f_{r,g}(\mathbf{x}^T \mathbf{g}) d\mathbf{x} \\ &= \int q_G(y_d, \mathbf{y}_{-d}) [\nabla_{\mathbf{G}^{-1} \mathbf{y}} \log \frac{p(\mathbf{G}^{-1} \mathbf{y})}{q(\mathbf{G}^{-1} \mathbf{y})}]^T \mathbf{r} f_{r,g}(y_d) d\mathbf{y}_{-d} dy_d. \end{aligned} \quad (27)$$

The next step is to choose a specific form for the test function  $f_{r,g}(y_d)$ . Define

$$f_{r,g}(y_d) = \int q_G(y_d, \mathbf{y}_{-d}) [\nabla_{\mathbf{G}^{-1} \mathbf{y}} \log \frac{p(\mathbf{G}^{-1} \mathbf{y})}{q(\mathbf{G}^{-1} \mathbf{y})}]^T \mathbf{r} d\mathbf{y}_{-d}. \quad (28)$$

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 6, this selected test function can be re-written into

$$\begin{aligned} & \int q_G(y_d, \mathbf{y}_{-d}) [\nabla_{\mathbf{G}^{-1} \mathbf{y}} \log \frac{p(\mathbf{G}^{-1} \mathbf{y})}{q(\mathbf{G}^{-1} \mathbf{y})}]^T \mathbf{r} d\mathbf{y}_{-d} = \int q(\mathbf{x}) [\nabla_{\mathbf{x}} \log \frac{p(\mathbf{x})}{q(\mathbf{x})}]^T \mathbf{r} \delta(y_d - \mathbf{x}^T \mathbf{g}) d\mathbf{x} \\ &= \mathcal{R}[I_{q,p}](y_d, \mathbf{g}). \end{aligned}$$

This is exactly the Radon transform of the function  $I_{q,p} = q(\mathbf{x})(\mathbf{s}_p(\mathbf{x}) - \mathbf{s}_q(\mathbf{x}))^T \mathbf{r}$ . 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  $q$  (Liu et al., 2016).

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

$$\begin{aligned} & \int q_G(y_d, \mathbf{y}_{-d}) [\nabla_{\mathbf{G}^{-1} \mathbf{y}} \log \frac{p(\mathbf{G}^{-1} \mathbf{y})}{q(\mathbf{G}^{-1} \mathbf{y})}]^T \mathbf{r} \int q_G(y_d, \mathbf{u}_{-d}) [\nabla_{\mathbf{G}^{-1} \mathbf{u}} \log \frac{p(\mathbf{G}^{-1} \mathbf{u})}{q(\mathbf{G}^{-1} \mathbf{u})}]^T \mathbf{r} d\mathbf{u}_{-d} d\mathbf{y}_{-d} dy_d \\ &= \int \left\{ \int q_G(y_d, \mathbf{y}_{-d}) [\nabla_{\mathbf{G}^{-1} \mathbf{y}} \log \frac{p(\mathbf{G}^{-1} \mathbf{y})}{q(\mathbf{G}^{-1} \mathbf{y})}]^T \mathbf{r} d\mathbf{y}_{-d} \right\} \\ & \quad \left\{ \int q_G(y_d, \mathbf{u}_{-d}) [\nabla_{\mathbf{G}^{-1} \mathbf{u}} \log \frac{p(\mathbf{G}^{-1} \mathbf{u})}{q(\mathbf{G}^{-1} \mathbf{u})}]^T \mathbf{r} d\mathbf{u}_{-d} \right\} dy_d \\ &= \int f_{r,g}^2(y_d) dy_d = \textcircled{1} \\ &\geq 0. \end{aligned} \quad (29)$$

Thus we have constructed a lower bound (Eq.(29)) for the supremum in Eq.(5) and it is greater than 0. Next, we show the expectation of this lower bound over  $p_g$  and  $p_r$  is 0 only if  $p = q$  a.e.. If so then Eq.(5) is 0 only if  $p = q$  a.e..

First, it is clearly that  $\textcircled{1} = 0$  iff.  $f_{r,g}(y_d) = 0$  a.e. By Lemma 6, we have  $f_{r,g}(y_d) = \mathcal{R}[I_{q,p}](y_d, \mathbf{g})$ . Thus, we have

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

where  $I_q = q(\mathbf{x})\mathbf{s}_q(\mathbf{x})^T \mathbf{r}$  and  $I_p = q(\mathbf{x})\mathbf{s}_p(\mathbf{x})^T \mathbf{r}$ .

Now we define the  $D$  dimensional Fourier transform operator  $\mathcal{F}_D$ , slice operator  $\mathcal{S}_1$  as in Theorem 3. Based on Fourier sliced lemma 3, we have

$$\begin{aligned} & \mathcal{R}[I_q](y_d, \mathbf{g}) = \mathcal{R}[I_p](y_d, \mathbf{g}) \\ & \Rightarrow \mathcal{F}_1[\mathcal{R}[I_q](y_d, \mathbf{g})] = \mathcal{F}_1[\mathcal{R}[I_p](y_d, \mathbf{g})] \\ & \Rightarrow \mathcal{S}_1[\mathcal{F}_D[I_q]](\cdot, \mathbf{g}) = \mathcal{S}_1[\mathcal{F}_D[I_p]](\cdot, \mathbf{g}). \end{aligned} \quad (30)$$

This means the one dimensional slice at direction  $\mathbf{g}$  for Fourier transform  $\mathcal{F}_D(I_q)$  and  $\mathcal{F}_D(I_p)$  are the same. Also note that the discrepancy (Eq.(5)) is defined by integrating over test directions  $\mathbf{g}$  with a uniform distribution  $p_g(\mathbf{g})$  over  $\mathbb{S}^{D-1}$ . This means if the discrepancy is zero, then Eq.(30) must hold true for  $\mathbf{g}$  a.e. over the hyper-sphere. Thus, we can show

$$\mathcal{F}_D(I_q) = \mathcal{F}_D(I_p) \quad \text{a.e.} \quad (31)$$

It is well-known that the Fourier transform is injective, thus, for any direction  $\mathbf{r}$ , we have

$$\begin{aligned}\mathcal{F}_D(I_q) &= \mathcal{F}_D(I_p) \\ \Rightarrow I_q &= I_p \\ \Rightarrow q(\mathbf{x})\mathbf{s}_q(\mathbf{x})^T \mathbf{r} &= q(\mathbf{x})\mathbf{s}_p(\mathbf{x})^T \mathbf{r} \\ \Rightarrow \mathbf{s}_q(\mathbf{x})^T \mathbf{r} &= \mathbf{s}_p(\mathbf{x})^T \mathbf{r}\end{aligned}\tag{32}$$

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

$$\mathbf{s}_q(\mathbf{x})^T \mathbf{r} = \mathbf{s}_p(\mathbf{x})^T \mathbf{r} \text{ for all } \mathbf{r} \Rightarrow \mathbf{s}_q(\mathbf{x}) = \mathbf{s}_p(\mathbf{x}) \Rightarrow p = q \text{ a.e.}$$

This finishes the proof of the ‘‘only if’’ part:  $\mathcal{S}(q, p) \geq 0$  and is 0 only if  $q = p$  a.e.  $\square$

### B.3 PROOF OF COROLLARY 1.1

To prove the corollary 1.1, we first propose a variant of SSD (Eq.(5)) by relaxing the score projection  $\mathbf{r}$ . We call it *orthogonal basis SSD*.

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

$$S_o(q, p) = \sum_{\mathbf{r} \in O_r} \int_{\mathbb{S}^{D-1}} p_g(\mathbf{g}) \sup_{f_{rg} \in \mathcal{F}_q} \mathbb{E}_q[\mathbf{s}_p^T(\mathbf{x}) f_{rg}(\mathbf{x}^T \mathbf{g}) + \mathbf{r}^T \mathbf{g} \nabla_{\mathbf{x}^T \mathbf{g}} f_{rg}(\mathbf{x}^T \mathbf{g})] d\mathbf{g} \tag{33}$$

is zero if and only if  $p = q$  a.e. One simple choice for  $O_r$  can be  $O_r = \{\mathbf{r}_1, \dots, \mathbf{r}_D\}$  where  $\mathbf{r}_d$  is one-hot vector with value 1 in  $d^{\text{th}}$  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  $O_r = \{\mathbf{r}_1, \dots, \mathbf{r}_D\}$  where  $\mathbf{r}_d$  is one-hot vector with value 1 in  $i^{\text{th}}$  component. For general  $O_r$ , we can simply apply a inverse rotation matrix  $\mathbf{R}^{-1}$  to recover this special case.

From Eq.(32), we have for direction  $\mathbf{r}_d$ ,

$$\begin{aligned}\mathbf{s}_q(\mathbf{x})^T \mathbf{r}_d &= \mathbf{s}_p(\mathbf{x})^T \mathbf{r}_d \\ \Rightarrow \nabla_{x_d} \log q(x_d, \mathbf{x}_{-d}) &= \nabla_{x_d} \log p(x_d, \mathbf{x}_{-d}) \\ \Rightarrow \nabla_{x_d} \log q(x_d | \mathbf{x}_{-d}) &= \nabla_{x_d} \log p(x_d | \mathbf{x}_{-d}).\end{aligned}$$

If the above holds true for all directions  $\mathbf{r}_d \in O_r$ , then the score of the complete conditional for  $q$  and  $p$  are equal. Then from Lemma 1 in (Singhal et al., 2019), we have  $p = q$  a.e.  $\square$

Now we can prove Corollary 1.1 using Remark 1.

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

$$\sup_{f_{rg} \in \mathcal{F}_q} \mathbb{E}_q[\mathbf{s}_p^T(\mathbf{x}) f_{rg}(\mathbf{x}^T \mathbf{g}) + \mathbf{r}^T \mathbf{g} \nabla_{\mathbf{x}^T \mathbf{g}} f_{rg}(\mathbf{x}^T \mathbf{g})] = 0$$

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  $S_{max}(q, p) = 0$  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 (Liu et al., 2016; Chwialkowski et al., 2016):

$$h_{p,r,g}(\mathbf{x}, \mathbf{y}) = \langle \xi_{p,r,g}(\mathbf{x}, \cdot), \xi_{p,r,g}(\mathbf{y}, \cdot) \rangle_{\mathcal{H}_{rg}}. \tag{34}$$

Next, we show that  $\xi_{p,r,g}(\mathbf{x}, \cdot)$  is Bochner integrable (Christmann & Steinwart), i.e.

$$\mathbb{E}_q \|\xi_{p,r,g}(\mathbf{x})\|_{\mathcal{H}_{rg}} \leq \sqrt{\mathbb{E}_q \|\xi_{p,r,g}(\mathbf{x})\|_{\mathcal{H}_{rg}}^2} = \sqrt{\mathbb{E}_q [h_{p,r,g}(\mathbf{x}, \mathbf{x})]} \leq \infty. \quad (35)$$

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))

$$\begin{aligned} & \left\| \sup_{f_{rg} \in \mathcal{H}_{rg}, \|f_{rg}\| \leq 1} \mathbb{E}_q [s_p^r(\mathbf{x}) f_{rg}(\mathbf{x}^T \mathbf{g}) + \mathbf{r}^T \mathbf{g} \nabla_{\mathbf{x}^T \mathbf{g}} f_{rg}(\mathbf{x}^T \mathbf{g})] \right\|^2 \\ &= \left\| \sup_{f_{rg} \in \mathcal{H}_{rg}, \|f_{rg}\| \leq 1} \mathbb{E}_q [\langle s_p^r(\mathbf{x}) k_{rg}(\mathbf{x}^T \mathbf{g}, \cdot) + \mathbf{r}^T \mathbf{g} \nabla_{\mathbf{x}^T \mathbf{g}} k_{rg}(\mathbf{x}^T \mathbf{g}, \cdot), f_{rg} \rangle_{\mathcal{H}_{rg}}] \right\|^2 \\ &= \left\| \sup_{f_{rg} \in \mathcal{H}_{rg}, \|f_{rg}\| \leq 1} \langle f_{rg}, \mathbb{E}_q [s_p^r(\mathbf{x})^T \mathbf{r} k_{rg}(\mathbf{x}^T \mathbf{g}, \cdot) + \mathbf{r}^T \mathbf{g} \nabla_{\mathbf{x}^T \mathbf{g}} k_{rg}(\mathbf{x}^T \mathbf{g}, \cdot)] \rangle_{\mathcal{H}_{rg}} \right\|^2 \\ &= \|\mathbb{E}_q [\xi_{p,r,g}(\mathbf{x})]\|_{\mathcal{H}_{rg}}^2 \\ &= \langle \mathbb{E}_q [\xi_{p,r,g}(\mathbf{x}, \cdot)], \mathbb{E}_q [\xi_{p,r,g}(\mathbf{x}', \cdot)] \rangle_{\mathcal{H}_{rg}} \\ &= \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim q} [h_{p,r,g}(\mathbf{x}, \mathbf{x}')]. \end{aligned} \quad (36)$$

□

## 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  $\mathbf{r}$  and  $\mathbf{g}$ . From Theorem 2, we have

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

From Assumption 5, we know  $k_{rg}(\mathbf{x}^T \mathbf{g}, \cdot)$  belongs to the Stein class of  $q$ . Then we follow the same proof technique in Proposition 1 but replace the test function  $f_{rg}(\mathbf{x}^T \mathbf{g})$  with  $k_{rg}(\mathbf{x}^T \mathbf{g}, \cdot)$ . This gives

$$\mathbb{E}_q [s_q^r(\mathbf{x})^T \mathbf{r} k_{rg}(\mathbf{x}^T \mathbf{g}, \cdot) + \mathbf{r}^T \mathbf{g} \nabla_{\mathbf{x}^T \mathbf{g}} k_{rg}(\mathbf{x}^T \mathbf{g}, \cdot)] = 0, \quad (37)$$

i.e.  $\mathbb{E}_q [\xi_{p,r,g}(\mathbf{x}, \cdot)] = 0$ . 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

$$\begin{aligned} D_{rg}(q, p) &= 0 \\ \Rightarrow \|\mathbb{E}_q [\xi_{p,r,g}(\mathbf{x})]\|_{\mathcal{H}_{rg}} &= 0 \\ \Rightarrow \mathbb{E}_q [\xi_{p,r,g}(\mathbf{x}, \cdot)] &= 0 \end{aligned}$$

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

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

Next, we denote  $\mathbf{G}$  as an arbitrary rotation with the  $d$ th entry as the test direction  $\mathbf{g}$ , and  $\mathbf{y} = \mathbf{G}\mathbf{x}$  with  $y_d = \mathbf{x}^T \mathbf{g}$ . Then from Lemma 4, we have

$$\begin{aligned} & \int q(\mathbf{x}) \nabla_{\mathbf{x}} \log \frac{p(\mathbf{x})}{q(\mathbf{x})}^T \mathbf{r} k_{rg}(\mathbf{x}^T \mathbf{g}, \cdot) d\mathbf{x} \\ &= \int q_G(y_d, \mathbf{y}_{-d}) \nabla_{\mathbf{G}^{-1}\mathbf{y}} \log \frac{q(\mathbf{G}^{-1}\mathbf{y})}{p(\mathbf{G}^{-1}\mathbf{y})}^T \mathbf{r} k_{rg}(y_d, \cdot) d\mathbf{y}_{-d} dy_d \\ &= \int q_G(y_d) k_{rg}(y_d, \cdot) \int q_G(\mathbf{y}_{-d} | y_d) \nabla_{\mathbf{G}^{-1}\mathbf{y}} \log \frac{q(\mathbf{G}^{-1}\mathbf{y})}{p(\mathbf{G}^{-1}\mathbf{y})}^T \mathbf{r} d\mathbf{y}_{-d} dy_d \\ &= \int q_G(y_d) k_{rg}(y_d, \cdot) H_r(y_d) dy_d \end{aligned}$$

where  $H_r(y_d) = \int q_G(\mathbf{y}_{-d} | y_d) \nabla_{\mathbf{G}^{-1}\mathbf{y}} \log \frac{q(\mathbf{G}^{-1}\mathbf{y})}{p(\mathbf{G}^{-1}\mathbf{y})}^T \mathbf{r} d\mathbf{y}_{-d}$ . 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 Carmeli et al. (2010), its embedding is zero if and only if  $H_r(\cdot) = 0$ . This implies

$$\begin{aligned} H_r(y_d) &= \int q_G(\mathbf{y}_{-d}|y_d) \nabla_{\mathbf{G}^{-1}\mathbf{y}} \log \frac{q(\mathbf{G}^{-1}\mathbf{y})}{p(\mathbf{G}^{-1}\mathbf{y})}^T \mathbf{r} d\mathbf{y}_{-d} = 0 \\ \Rightarrow \int q_G(y_d, \mathbf{y}_{-d}) \nabla_{\mathbf{G}^{-1}\mathbf{y}} \log \frac{q(\mathbf{G}^{-1}\mathbf{y})}{p(\mathbf{G}^{-1}\mathbf{y})}^T \mathbf{r} d\mathbf{y}_{-d} &= 0 \\ \Rightarrow \int q(\mathbf{x})(s_p^r(\mathbf{x}) - s_q^r(\mathbf{x})) \delta(y_d - \mathbf{x}^T \mathbf{g}) d\mathbf{x} &= 0 \end{aligned}$$

where the third equality is from Lemma 6. 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.  $\square$

## D DEEP KERNEL

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 SKSD or maxSKSD would result in a kernel evaluated on  $\phi(\mathbf{x}^T \mathbf{g})$ , which is less desirable. To make the kernel evaluated on the transformed input  $\phi(\mathbf{x})^T \mathbf{g}$ , we need to slightly adapt the form of SKSD (Eq.10). Assume we have a smooth injective mapping  $\phi$ , we define the following term

$$\xi_{p,r,g,\phi}(\mathbf{x}, \cdot) = s_p^r(\mathbf{x}) k_{rg}(\phi^g(\mathbf{x}), \cdot) + C_\phi(\mathbf{x}) \nabla_{\phi^g(\mathbf{x})} k_{rg}(\phi^g(\mathbf{x}), \cdot) \quad (38)$$

and

$$\begin{aligned} h_{p,r,g,\phi}(\mathbf{x}, \mathbf{y}) &= \\ &s_p^r(\mathbf{x}) k_{rg}(\phi^g(\mathbf{x}), \phi^g(\mathbf{y})) s_p^r(\mathbf{y}) + C_{r,g,\phi}(\mathbf{x}) s_p^r(\mathbf{y}) \nabla_{\phi^g(\mathbf{x})} k_{rg}(\phi^g(\mathbf{x}), \phi^g(\mathbf{y})) \\ &+ C_{r,g,\phi}(\mathbf{y}) s_p^r(\mathbf{x}) \nabla_{\phi^g(\mathbf{y})} k_{rg}(\phi^g(\mathbf{x}), \phi^g(\mathbf{y})) \\ &+ C_{r,g,\phi}(\mathbf{x}) C_{r,g,\phi}(\mathbf{y}) \nabla_{\phi^g(\mathbf{x}), \phi^g(\mathbf{y})}^2 k_{rg}(\phi^g(\mathbf{x}), \phi^g(\mathbf{y})) \end{aligned} \quad (39)$$

where  $C_{r,g,\phi}(\mathbf{x}) = \mathbf{r}^T \frac{\partial \phi(\mathbf{x})}{\partial \mathbf{x}} \mathbf{g}$  and  $\phi^g(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{g}$ . We provide the following theorem to prove the validity of the corresponding SKSD discrepancy measure.

**Theorem 4.** (Deep Kernel SKSD) For two probability distributions  $p$  and  $q$ , assume we have a smooth injective mapping  $\phi(\mathbf{x})$ , such that Assumptions 1, 2 and 5 are satisfied, and  $\mathbb{E}_q[h_{p,r,g,\phi}(\mathbf{x}, \mathbf{x})] < \infty$  for all  $\mathbf{r}$  and  $\mathbf{g}$ , then we propose deep kernel SKSD (Deep-SKSD) as

$$DSK_o(q, p) = \sum_{\mathbf{r} \in \mathcal{O}_r} \int_{\mathbb{S}^{D-1}} p_g(\mathbf{g}) D_{r,g,\phi}^2(q, p) d\mathbf{g}, \quad D_{r,g,\phi}^2(q, p) = \mathbb{E}_q[h_{p,r,g,\phi}(\mathbf{x}, \mathbf{x}')], \quad (40)$$

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.(10)). 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

$$h_{p,r,g,\phi}(\mathbf{x}, \mathbf{y}) = \langle \xi_{p,r,g,\phi}(\mathbf{x}, \cdot), \xi_{p,r,g,\phi}(\mathbf{y}, \cdot) \rangle_{\mathcal{H}_{rg}}.$$

By Assumption 5,  $k_{rg}$  belongs to the Stein class of  $q$ . Thus,  $k_{rg}(\phi^g(\mathbf{x}), \cdot)$  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

$$F(\mathbf{x}) = \begin{bmatrix} r_1 k_{rg}(\phi^g(\mathbf{x}), \cdot) \\ r_2 k_{rg}(\phi^g(\mathbf{x}), \cdot) \\ \vdots \\ r_D k_{rg}(\phi^g(\mathbf{x}), \cdot) \end{bmatrix}$$

and substitute this into Stein identity. This returns

$$\begin{aligned} & \int q(\mathbf{x})[s_q^T(\mathbf{x})k_{rg}(\phi^g(\mathbf{x}), \cdot) + C_{r,g,\phi}(\mathbf{x})\nabla_{\phi^g(\mathbf{x})}k_{rg}(\phi^g(\mathbf{x}), \cdot)]d\mathbf{x} = 0 \\ \Rightarrow & \mathbb{E}_q[\xi_{q,r,g,\phi}(\mathbf{x}, \cdot)] = 0 \\ \Rightarrow & D_{r,g,\phi}^2(q, q) = \mathbb{E}_q[h_{q,r,g,\phi}(\mathbf{x}, \mathbf{x}')] = \langle \mathbb{E}_q[\xi_{q,r,g,\phi}(\mathbf{x}, \cdot)], \mathbb{E}_q[\xi_{q,r,g,\phi}(\mathbf{x}', \cdot)] \rangle_{\mathcal{H}_{rg}} = 0. \end{aligned}$$

Therefore, if  $p = q$  a.e., then  $DSK_o(q, p) = 0$ .

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

$$DSK_o(q, p) = 0 \quad \Rightarrow \quad \mathbb{E}_q[\xi_{p,r,g,\phi}(\mathbf{x}, \cdot)] = 0.$$

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

$$\mathbb{E}_q[\xi_{p,r,g,\phi}(\mathbf{x}, \cdot)] = \mathbb{E}_q[(s_p^T(\mathbf{x}) - s_q^T(\mathbf{x}))k_{rg}(\phi^g(\mathbf{x}), \cdot)].$$

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

$$\begin{aligned} q_\phi(\mathbf{y}) &= q(\mathbf{x})|\mathbf{J}|^{-1}, \\ q_{G\phi}(\mathbf{u}) &= q_\phi(\mathbf{y})|\mathbf{G}|^{-1} = q_\phi(\mathbf{y}), \end{aligned} \tag{41}$$

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

$$\begin{aligned} & \int q(\mathbf{x})[\nabla_{\mathbf{x}} \log \frac{p(\mathbf{x})}{q(\mathbf{x})}]^T \mathbf{r} k_{rg}(\phi^g(\mathbf{x}), \cdot) d\mathbf{x} \\ &= \int q_\phi(\mathbf{y})|\mathbf{J}|[\nabla_{\phi^{-1}(\mathbf{y})} \log \frac{p(\phi^{-1}(\mathbf{y}))}{q(\phi^{-1}(\mathbf{y}))}]^T \mathbf{r} k_{rg}(\mathbf{y}^T \mathbf{g}, \cdot) |\mathbf{J}|^{-1} d\mathbf{y} \\ &= \int q_{G\phi}(u_d, \mathbf{u}_{-d})[\nabla_{\phi^{-1}(\mathbf{G}^{-1}\mathbf{u})} \log \frac{p(\phi^{-1}(\mathbf{G}^{-1}\mathbf{u}))}{q(\phi^{-1}(\mathbf{G}^{-1}\mathbf{u}))}]^T \mathbf{r} k_{rg}(u_d, \cdot) |\mathbf{G}^{-1}| du_d d\mathbf{u}_{-d} \\ &= \int q_{G\phi}(u_d) k_{rg}(u_d, \cdot) \int q_{G\phi}(\mathbf{u}_{-d} | u_d) [\nabla_{\phi^{-1}(\mathbf{G}^{-1}\mathbf{u})} \log \frac{p(\phi^{-1}(\mathbf{G}^{-1}\mathbf{u}))}{q(\phi^{-1}(\mathbf{G}^{-1}\mathbf{u}))}]^T \mathbf{r} d\mathbf{u}_{-d} du_d \\ &= \int q_{G\phi}(u_d) k_{rg}(u_d, \cdot) H(u_d) du_d. \end{aligned}$$

Following the proof steps in Theorem 3, we have  $H(u_d) = 0$ . Then by Lemma 6, we have

$$\begin{aligned} & \int q_{G\phi}(\mathbf{u}_{-d}, u_d) [\nabla_{\phi^{-1}(\mathbf{G}^{-1}\mathbf{u})} \log \frac{p(\phi^{-1}(\mathbf{G}^{-1}\mathbf{u}))}{q(\phi^{-1}(\mathbf{G}^{-1}\mathbf{u}))}]^T \mathbf{r} d\mathbf{u}_{-d} \\ &= \int q_\phi(\mathbf{y}) [\nabla_{\phi^{-1}(\mathbf{y})} \log \frac{p(\phi^{-1}(\mathbf{y}))}{q(\phi^{-1}(\mathbf{y}))}]^T \mathbf{r} \delta(u_d - \mathbf{y}^T \mathbf{g}) d\mathbf{y}. \end{aligned}$$

Finally using similar proof techniques in Theorem 1, we have

$$p(\phi^{-1}(\mathbf{y})) = q(\phi^{-1}(\mathbf{y})).$$

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

## E CLOSED-FORM SOLUTIONS FOR $\mathbf{G}$

In general, such closed-form solutions of  $\mathbf{G}$  is difficult to find, and we have to resort to gradient-based optimization for such task. However, the closed-form solutions exists under certain conditions. In the following, we give the closed-form solution of  $\mathbf{G}$  under conditions that  $p$  and  $q$  are full-factorized. Let define two distributions  $p, q$  with support  $\mathbb{R}^D$ , such that  $\log p(\mathbf{x}) = \sum_{d=1}^D \log p_d(x_d)$  and  $\log q(\mathbf{x}) = \sum_{d=1}^D \log q_d(x_d)$ . Now we consider the  $\max SSD$ -g (Eq.6) with  $\mathcal{O}_r$  to be a group of one-hot vectors. Thus, eq.6 becomes

$$S_{\max}(q, p) = \sum_{d=1}^D \max_{f_d \in \mathcal{F}_q, \mathbf{g}_d \in \mathbb{S}^{D-1}} \mathbb{E}_q[S_{p,d}(x_d) f_d(\mathbf{x}^T \mathbf{g}_d) + g_{d,d} \nabla_{\mathbf{x}^T \mathbf{g}_d} f_d(\mathbf{x}^T \mathbf{g}_d)] \tag{42}$$

where  $S_{p,d}(x_d) = \nabla_{x_d} \log p_d(x_d)$  and  $g_{d,d}$  is the  $d^{\text{th}}$  element of  $\mathbf{g}_d$ . It is difficult to directly solve this optimization. Instead, we can find its upper bound, and show that such upper bound can be recovered by choosing a specific form of  $\mathbf{G}$ . Thus, such  $\mathbf{G}$  will be the optimal one.

Let's consider the Stein divergence with test function  $H(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^D$  and  $H(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_D(\mathbf{x})]^T$ . We have

$$\begin{aligned} SD(q, p) &= \max_{H \in \mathcal{F}_q} \mathbb{E}_q[\nabla_{\mathbf{x}} \log p(\mathbf{x})^T H(\mathbf{x}) + \nabla_{\mathbf{x}}^T H(\mathbf{x})] \\ &= \max_{H \in \mathcal{F}_q} \mathbb{E}_q\left[\sum_{d=1}^D \nabla_{x_d} \log p_d(x_d) h_d(\mathbf{x}) + \nabla_{x_d} h_d(\mathbf{x})\right] \\ &= \max_{H \in \mathcal{F}_q} \sum_{d=1}^D \mathbb{E}_q[S_{p,d}(x_d) h_d(\mathbf{x}) + \nabla_{x_d} h_d(\mathbf{x})] \\ &= \sum_{d=1}^D \max_{h_d \in \mathcal{F}_q} \mathbb{E}_q[S_{p,d}(x_d) h_d(\mathbf{x}) + \nabla_{x_d} h_d(\mathbf{x})] \end{aligned} \quad (43)$$

where  $S_{p,d}(x_d) = \nabla_{x_d} \log p_d(x_d)$ . It is trivial that  $SD(q, p) \geq S_{max}(q, p)$  because  $h_d(\mathbf{x})$  is less restrictive than  $f_d(\mathbf{x}^T \mathbf{g}_d)$ . From [Hu et al. \(2018\)](#), we can obtain the optimal form for  $H^*(\mathbf{x}) \propto S_p(\mathbf{x}) - S_q(\mathbf{x})$  where  $S_p(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})$ . Therefore,  $h_d^*(\mathbf{x}) = h_d^*(x_d) \propto S_{p,d}(x_d) - S_{q,d}(x_d)$ . By substitution into eq.43, we have

$$SD(q, p) = \sum_{d=1}^D \mathbb{E}_q[S_{p,d}(x_d) h_d^*(x_d) + \nabla_{x_d} h_d^*(x_d)] \quad (44)$$

We note that eq.44 can be recovered by  $\max SSD$ -g (eq.42) with  $\mathbf{g}_d = [0, \dots, 1_d, \dots, 0]^T$  where  $d^{\text{th}}$  element  $1_d = 1$ , and  $f_d(x_d) \propto S_{p,d}(x_d) - S_{q,d}(x_d)$ . Thus, the optimal  $\mathbf{G} = \mathbf{I}$ , which is an identity matrix.

## F APPLICATIONS OF MAXSKSD

### F.1 GOODNESS-OF-FIT TEST

We propose a Goodness-of-fit test method based on the U-statistics of maxSKSD (Eq.(12)) given the optimal test direction  $\mathbf{g}_r$ . 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  $\widehat{SK}_{max}(q, p)$  is asymptotically normal. Particularly,

$$\sqrt{N}(\widehat{SK}_{max}(q, p) - SK_{max}(q, p)) \xrightarrow{d} \mathcal{N}(0, \sigma_h^2) \quad (45)$$

where  $\sigma_h^2 = \text{var}_{\mathbf{x} \sim q}(\sum_{\mathbf{r} \in O_r} \mathbb{E}_{\mathbf{x}' \sim q}[h_{p,r,g_r}(\mathbf{x}, \mathbf{x}')])$  and  $\sigma_h \neq 0$

2. If  $q = p$ , we have a degenerated U-statistics with  $\sigma_h = 0$  and

$$N\widehat{SK}_{max}(q, p) \xrightarrow{d} \sum_{j=1}^{\infty} c_j (Z_j^2 - 1) \quad (46)$$

where  $\{Z_j\}$  are i.i.d standard Gaussian variables, and  $\{c_j\}$  are the eigenvalues of the kernel  $\sum_{\mathbf{r} \in O_r} h_{p,r,g_r}(\mathbf{x}, \mathbf{x}')$  under  $q(\mathbf{x})$ . In other words, they are the solutions of  $c_j \phi_j(\mathbf{x}) = \int_{\mathbf{x}'} \sum_{\mathbf{r} \in O_r} h_{p,r,g_r}(\mathbf{x}, \mathbf{x}') \phi_j(\mathbf{x}') q(\mathbf{x}') d\mathbf{x}'$ .

*Proof.* We can directly use the results in Section 5.5 of ([Serfling, 2009](#)). 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  $\mathbb{E}_{\mathbf{x}' \sim q}[h_{p,r,g_r}]$  as

$$\mathbb{E}_{\mathbf{x}' \sim q}[h_{p,r,g_r}(\mathbf{x}, \mathbf{x}')] = \langle \xi_{p,r,g_r}(\mathbf{x}, \cdot), \mathbb{E}_{\mathbf{x}' \sim q}[\xi_{p,r,g_r}(\mathbf{x}', \cdot)] \rangle_{\mathcal{H}_{r,g_r}}$$

From the Eq.(37) in theorem 3, we have  $\mathbb{E}_{\mathbf{x}' \sim q}[\xi_{p,r,g}(\mathbf{x}', \cdot)] = 0$ . Thus,  $\mathbb{E}_{\mathbf{x}' \sim q}[h_{p,r,g_r}] = 0$  for all  $\mathbf{r} \in O_r$ . Thus, we have  $\sigma_h = 0$  when  $q = p$ .

We assume when  $p \neq q$ ,  $\sigma_h = 0$ . This means  $\mathbb{E}_{\mathbf{x}' \sim q}[h_{p,r,g}(\mathbf{x}, \mathbf{x}')] = c_r$  where  $c_r$  is a constant. Thus,

$$\begin{aligned} c_r &= \mathbb{E}_{\mathbf{x} \sim p}[\mathbb{E}_{\mathbf{x}' \sim q}[h_{p,r,g}(\mathbf{x}, \mathbf{x}')] ] \\ \Rightarrow c_r &= \mathbb{E}_{\mathbf{x}' \sim q}[\mathbb{E}_{\mathbf{x} \sim p}[h_{p,r,g}(\mathbf{x}, \mathbf{x}')] ] \end{aligned}$$

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

This theorem indicates a well-defined limit distribution for maxSKSD U-statistics. Next, similar to the previous work (Liu et al., 2016), we adopt the bootstrap method (Arcones & Gine, 1992; Huskova & Janssen, 1993). The quantile computed by the bootstrap samples (Eq.13) is consistent to the one using degenerated U-statistics. This consistence is established in (Huskova & Janssen, 1993; Arcones & Gine, 1992).

## F.2 SLICED SVGD

First, we introduce one result from Liu & Wang (2016), that shows the connections between the SD and KL divergence between the particle's underlying distribution  $q$  and target  $p$ .

**Lemma 7.** (Liu & Wang, 2016) Let  $T(\mathbf{x}) = \mathbf{x} + \epsilon\phi(\mathbf{x})$  and  $q_{[T]}(\mathbf{z})$  be the density of  $\mathbf{z} = T(\mathbf{x})$  when  $\mathbf{x} \sim q(\mathbf{x})$ . With  $\mathcal{A}_p$  the Stein operator defined in Eq.(1), we have

$$\nabla_{\epsilon} KL[q_{[T]}||p]|_{\epsilon=0} = -\mathbb{E}_q[\mathcal{A}_p\phi(\mathbf{x})]. \quad (47)$$

To derive the sliced version of SVGD, we follow the similar recipe of Liu & Wang (2016) by first connecting the SSD with KL divergence (like Lemma 7), and then derive the optimal perturbation directions (like Lemma 1 in background section 2.2). To achieve this, we modify the flow mapping to  $T_G(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^D$  as  $T_G(\mathbf{x}) = \mathbf{x} + \epsilon\phi_G(\mathbf{x})$ . Specifically for  $\phi_G(\mathbf{x})$ , we adopt  $D$  univariate perturbations instead of one multivariate perturbation:

$$\phi_G(\mathbf{x}) = \begin{bmatrix} \phi_{g_1}(\mathbf{x}^T \mathbf{g}_1) \\ \vdots \\ \phi_{g_D}(\mathbf{x}^T \mathbf{g}_D) \end{bmatrix} \quad (48)$$

where  $G = [\mathbf{g}_1, \dots, \mathbf{g}_D] \in \mathbb{R}^{D \times D}$  represents slice matrix. For this specific mapping we have the following result analogous to Lemma 7.

**Lemma 8.** Let  $T_G(\mathbf{x}) = \mathbf{x} + \epsilon\phi_G(\mathbf{x})$  where  $\phi_G$  is defined as Eq.(48). Define  $q_{[T_G]}(\mathbf{z})$  as the density of  $\mathbf{z} = T_G(\mathbf{x})$  when  $\mathbf{x} \sim q(\mathbf{x})$ , with slice matrix  $\mathbf{G}$ , we have

$$\nabla_{\epsilon} [q_{[T_G]}||p]|_{\epsilon=0} = -\sum_{d=1}^D \mathbb{E}_q[s_p^d(\mathbf{x})\phi_{g_d}(\mathbf{x}^T \mathbf{g}_d) + g_{d,d}\nabla_{\mathbf{x}^T \mathbf{g}_d}\phi_{g_d}(\mathbf{x}^T \mathbf{g}_d)] \quad (49)$$

where  $s_p^d(\mathbf{x}) = \nabla_{x_d} \log p(\mathbf{x})$  and  $g_{d,d}$  is the  $d^{\text{th}}$  element in  $\mathbf{g}_d$ .

*Proof.* This can be easily verified by substituting Eq.(48) into Eq.(47).  $\square$

Eq.(49) is similar to maxSSD (Eq.(6)) where the optimal test directions and test functions are replaced with matrix  $\mathbf{G}$  and perturbation  $\phi_{g_d}(\mathbf{x})$ .  $O_r$  takes the values one-hot vectors. The main difference between this decrease magnitude and maxSSD is that we do not assume  $\mathbf{G}$  is optimal. Next, we show how to obtain an analytic descent directions that maximize the decrease magnitude.

By restricting each perturbation  $\phi_{g_d} \in \mathcal{H}_{rg_d}$  where  $\mathcal{H}_{rg_d}$  is an RKHS equipped with kernel, we have the following result.

**Lemma 9.** Assume the conditions in lemma 8. If for each perturbation  $\phi_{g_d} \in \mathcal{H}_{r_{g_d}}$  where  $\mathcal{H}_{r_{g_d}}$  is an RKHS equipped with kernel  $k_{r_{g_d}}$  and  $\|\phi_{g_d}\|_{\mathcal{H}_{r_{g_d}}} \leq D_{r_{g_d}}(q, p)$ , then the steepest descent direction for  $d^{\text{th}}$  perturbation is

$$\phi_{g_d}^*(\cdot) = \mathbb{E}_q[\xi_{p,r_d,g_d}(\mathbf{x}, \cdot)], \quad (50)$$

and

$$\nabla_{\epsilon} KL[q_{[T_G]}||p]|_{\epsilon=0} = - \sum_{d=1}^D D_{d_{g_d}}^2(q, p), \quad (51)$$

where  $D_{d_{g_d}}^2(q, p) = \mathbb{E}_q[h_{p,r_d,g_d}(\mathbf{x}, \mathbf{x}')] \text{ with one-hot vector } \mathbf{r}_d$ .

*Proof.* We show this result using the reproducing property of RKHS  $\mathcal{H}_{r_{g_d}}$ . The supremum of Eq.(49) can be re-written as

$$\begin{aligned} & \sup_{\phi_G} \sum_{d=1}^D \mathbb{E}_q[s_p^d(\mathbf{x})\phi_{g_d}(\mathbf{x}^T \mathbf{g}_d) + \mathbf{r}_d^T \mathbf{g}_d \nabla_{\mathbf{x}^T \mathbf{g}_d} \phi_{g_d}(\mathbf{x}^T \mathbf{g}_d)] \\ &= \sum_{d=1}^D \sup_{\phi_{g_d}} \mathbb{E}_q[s_p^d(\mathbf{x})\phi_{g_d}(\mathbf{x}^T \mathbf{g}_d) + \mathbf{r}_d^T \mathbf{g}_d \nabla_{\mathbf{x}^T \mathbf{g}_d} \phi_{g_d}(\mathbf{x}^T \mathbf{g}_d)] \\ &= \sum_{d=1}^D \sup_{\substack{\phi_{g_d} \in \mathcal{H}_{r_{g_d}} \\ \|\phi_{g_d}\|_{\mathcal{H}_{r_{g_d}}} \leq D_{r_{g_d}}(q,p)}} \mathbb{E}_q[\langle s_p^d(\mathbf{x})k_{r_{g_d}}(\mathbf{x}^T \mathbf{g}_d, \cdot) + \mathbf{r}_d^T \mathbf{g}_d \nabla_{\mathbf{x}^T \mathbf{g}_d} k_{r_{g_d}}(\mathbf{x}^T \mathbf{g}_d, \cdot), \phi_{g_d} \rangle_{\mathcal{H}_{r_{g_d}}}] \quad (52) \\ &= \sum_{d=1}^D \sup_{\substack{\phi_{g_d} \in \mathcal{H}_{r_{g_d}} \\ \|\phi_{g_d}\|_{\mathcal{H}_{r_{g_d}}} \leq D_{r_{g_d}}(q,p)}} \langle \mathbb{E}_q[\xi_{p,r_d,g_d}(\mathbf{x}, \cdot)], \phi_{g_d} \rangle_{\mathcal{H}_{r_{g_d}}} \\ &= \sum_{d=1}^D \mathbb{E}_q[h_{p,r_d,g_d}(\mathbf{x}, \mathbf{x}')] = \sum_{d=1}^D D_{d_{g_d}}^2(q, p), \end{aligned}$$

where the third equality is because of the Bochner integrability of  $\xi_{p,r_d,g_d}(\mathbf{x}, \cdot)$  shown in Theorem 2. And the optimal perturbation for  $d^{\text{th}}$  dimension is

$$\phi_{g_d}^*(\cdot) = \mathbb{E}_q[\xi_{p,r_d,g_d}(\mathbf{x}, \cdot)]. \quad (53)$$

□

Note that in Lemmas 8 and 9, we assume an arbitrary projection matrix  $\mathbf{G}$ . To find the steepest descent direction, one can maximize Eq.(51) w.r.t.  $\mathbf{G}$ . In this case this decrease magnitude Eq.(51) is identical to maxSKSD Eq.(11) with orthogonal basis  $O_r$  and optimal test directions  $\mathbf{G}$ .

The name *sliced* SVGD comes from that for each perturbation  $\phi_{g_d}(\cdot)$ , the kernel  $k_{r_{g_d}}$  and the repulsive force  $\mathbf{r}_d^T \mathbf{g}_d \nabla_{\mathbf{x}^T \mathbf{g}_d} k_{r_{g_d}}(\mathbf{x}^T \mathbf{g}_d, \cdot)$  are evaluated on  $\mathbf{x}^T \mathbf{g}_d$  instead of  $\mathbf{x}$  in SVGD. Although S-SVGd only uses one-dimensional projection of  $\mathbf{x}$ , it is still a valid inference method as long as the optimality of  $\mathbf{G}$  is ensured, because maxSKSD is a valid discrepancy measure.<sup>1</sup> The S-SVGd method is summarised in Algorithm 2. In practice this algorithm may violate the optimality condition of  $\mathbf{G}$  (due to estimation error using finite samples and local optimum found by

<sup>1</sup>Note that maximizing Eq.(51) w.r.t. sliced matrix  $\mathbf{G}$  is necessary, otherwise Eq.(51) is not a valid discrepancy measure, and a zero value does not imply  $p = q$ . In such case the resulting particle inference method is not asymptotically exact.

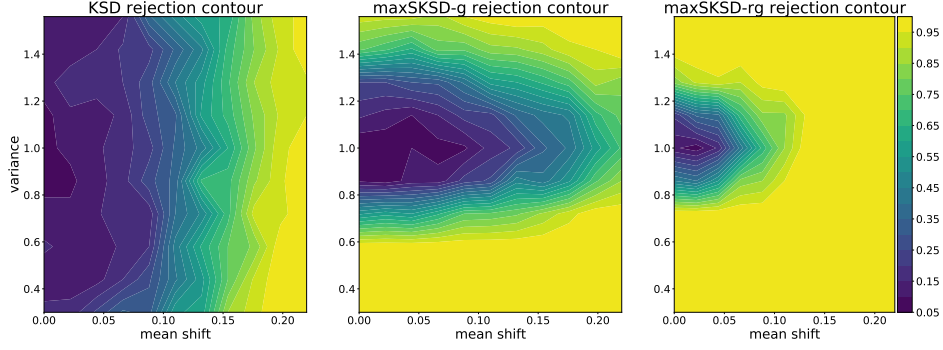


Figure 5: This rejection rate contour demonstrates the sensitivity of each GOF method to the change of mean/variance of two Gaussians under SDS. **(Left)** The rejection sensitivity of KSD. **(Middle)** Sensitivity of *maxSKSD-g*. **(Right)** Sensitivity of *maxSKSD-rg*.

gradient-based optimization), which is a common issue in many adversarial training procedure.

---

**Algorithm 2:** S-SVGD for variational inference

---

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

**Output** : Set of particles  $\{\mathbf{x}_i\}_{i=1}^N$  that approximates  $p$

**for**  $l \leq L$  **do**

    Update each particles  $\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \epsilon \phi_G^*(\mathbf{x}_i^l)$  where  $\phi_G^*(\mathbf{x}_i^l)$  is computed using Eq.(50);

    Find the optimal slice matrix  $\mathbf{G}$  by maximizing Eq.(51) using  $\{\mathbf{x}_i^{l+1}\}_{i=1}^N$

**end**

---

## G LIMITATIONS OF MAXSKSD-G

In this section, we discuss the potential limitations of *maxSKSD-g* and motivate the use of *maxSKSD-rg* to address such issue. We begin this journey by proposing a sensitivity test on the rejection rate between two Gaussians with different mean and variances. Then, we analyze the issue of *maxSKSD-g*, and why *maxSKSD-rg* can potentially address such pathology. In the end, we empirically show *maxSKSD-rg* indeed solves the problem under the same sensitivity test, and achieves the best performance.

### G.1 SENSITIVITY TEST OF TWO GAUSSIANS

Assume we have two fully factorized 50 dimensional Gaussians  $p, q$ , we propose to test the sensitivity of the GOF test method to the change of mean and variance of  $q$ . In the following experiment, we only change the mean and variance of the 1<sup>st</sup> dimension and keep the remaining 49 dimensions to be the same as  $p$ . We call this scheme *single dimension shift* (SDS). We include KSD as the baseline for comparison.

From the left and middle panel of figure.5, we notice that KSD is sensitive to the mean shift but less sensitive to the variance change. Specifically, KSD successfully detects the difference between  $q$  and  $p$  when the mean shift is beyond 0.18 but fails if the only changed term is the variance. On the other hand, *maxSKSD-g* is capable of detecting small variance changes but less sensitive to the mean change compared to KSD. This is consistent with the conclusion in section 4.1.1 that *maxSKSD-g* obtains nearly optimal rejection rate in Gaussian diffusion test where KSD completely fails. However, this still shows the potential limitations of *maxSKSD-g* that it may perform worse than the baselines under certain circumstances.

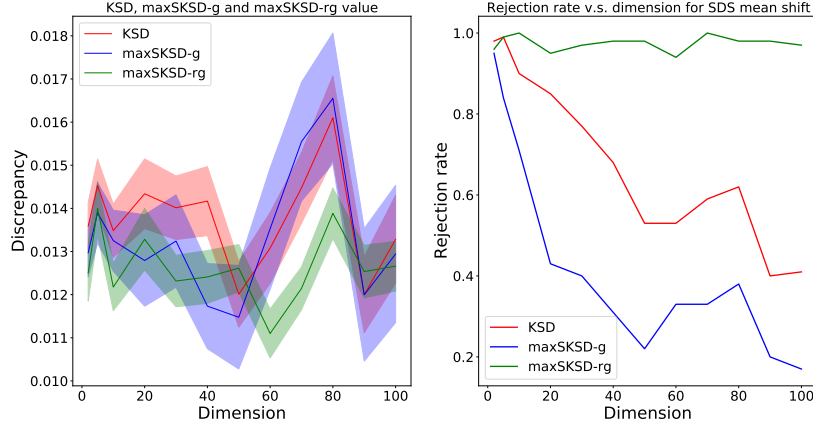


Figure 6: (Left) The discrepancy value of KSD, maxSKSD-g and maxSKSD-rg with standard error for SDS mean shift. Each plot is obtained by averaging 100 runs. (Right) The rejection rate for SDS mean shift with different dimensions.

## G.2 ANALYSIS OF THE PATHOLOGY

In this section, we give a detailed analysis on the potential reasons behind such limitation. From the setup of SDS mean shift, we know the mean of  $q$ ,  $\mu_q$ , differs from  $\mu_p$  only in the 1<sup>st</sup> dimension. We write down  $q_d(x_d)$  to be the marginal distribution for dimension  $d$ , and  $p_d$  accordingly. *maxSKSD-g* (Eq. 11) requires the orthogonal basis  $O_r$  and corresponding optimal slice matrix  $G$ . We set  $O_r$  to be the standard basis of  $\mathbb{R}^D$ , i.e. a set of one-hot vectors. Due to the fully factorized property of  $p$ ,  $q$  and SDS setup, the optimal slice matrix  $G$  is an identity matrix. Therefore, we can re-write the *maxSKSD-g* into a summation over a set of KSD between their marginals  $q_d$  and  $p_d$ . We first inspect the optimal test function in Eq. (7) for dimension  $d$ . Assume we use the same kernel  $k$  for all  $r, g$ , i.e.  $k = k_{rg}$ , we have

$$\begin{aligned}\xi_{p,r,g}(\mathbf{x}, \cdot) &= s_p^r(\mathbf{x})k(\mathbf{x}^T \mathbf{g}_r, \cdot) + \mathbf{r}^T \mathbf{g}_r \nabla_{\mathbf{x}^T \mathbf{g}_r} k(\mathbf{x}^T \mathbf{g}_r, \cdot) \\ &= s_{p_d}(x_d)k(x_d, \cdot) + \nabla_{x_d} k(x_d, \cdot)\end{aligned}\quad (54)$$

Substituting it into Eq. (9), we obtain

$$\begin{aligned}D_{rg}(q, p) &= \|\mathbb{E}_q[\xi_{p,r,g}(\mathbf{x})]\|_{\mathcal{H}_{rg}}^2 = \|\mathbb{E}_{q_d}[\xi_{p,r,g}(x_d)]\|_{\mathcal{H}_{rg}}^2 \\ &= \mathbb{E}_{q_d(x_d)q_d(x'_d)}[u_p(x_d, x'_d)] \\ &= D^2(q_d, p_d)\end{aligned}\quad (55)$$

This is exactly the KSD between the marginal  $q_d$  and  $p_d$ . Therefore, the *maxSKSD-g* is written as

$$\begin{aligned}SK_{max}(q, p) &= \sum_{\mathbf{r} \in O_r} \sup_{\mathbf{g}_r} D_{rg}^2(q, p) \\ &= \sum_{d=1}^D D_{rg}(q, p) \\ &= \sum_{d=1}^D D^2(q_d, p_d)\end{aligned}\quad (56)$$

This is the summation of the KSD between their marginals across all dimensions.

As the mean only differs in the first dimension, the dominant value for *maxSKSD-g* is the KSD between the first marginal,  $q_1$  and  $p_1$ . However, in practice, variances exist in other KSD terms inside the summation of *maxSKSD-g* and they provide nothing but noise. Therefore the overall variance of *maxSKSD-g* increases with the dimensions. For a high dimensional problem, the important signal can be easily buried due to the increasing variance. Figure 6 (Left) shows the discrepancy value

against dimensions. We can easily observe that the variance of  $\max\text{SKSD-g}$  increases with the dimensions. This is consistent with the above analysis. The variance of KSD also increases but is less obvious compared to  $\max\text{SKSD-g}$ . Figure 6 (Right) shows that  $\max\text{SKSD-g}$  also suffers from curse-of-dimensionality under the SDS mean shift due to such increasing variance.

### G.3 MAXSKSD-RG

Based on the above analysis, the underlying reason behind  $\max\text{SKSD-g}$  pathology is the noise provided by the irrelevant dimensions. This is the fundamental problem caused by choosing a orthogonal basis  $O_r$  for  $\max\text{SKSD}$ . In fact, for any orthogonal basis  $O_r$ , we can always create a such pathology where  $\max\text{SKSD-g}$  suffers from the extra noise provided by irrelevant basis. One potential way to address such problem is to introduce a metric to avoid such irrelevant basis. In here, we choose  $\max\text{SKSD}$  itself as the metric. Thus, instead of selecting a orthogonal basis  $O_r$ , we try to optimize the projections  $O_r \in \mathbb{R}^{D \times D}$  (not necessarily orthogonal). We write  $\max\text{SKSD-rg}$  as

$$SK_{\max-rg}(q, p) = \sup_{O_r, \mathbf{G}} \sum_{\mathbf{r} \in O_r} D_{rg_r}^2(q, p) \quad (57)$$

We test the performance of  $\max\text{SKSD-rg}$  in the same SDS mean shift setup. Due to the fully factorized property of two Gaussians, the optimal  $O_r$  consists of repeated one-hot vector  $\mathbf{r} = [1, 0, 0, \dots]$ . The optimal slice matrix  $\mathbf{G}$  share the same structure as  $O_r$ . Thus,  $\max\text{SKSD-rg}$  (Eq.57) becomes

$$SK_{\max-rg}(q, p) = D \times D^2(q_1, p_1) \quad (58)$$

which is  $D$  times KSD value between the marginals of the 1<sup>st</sup> dimensions. We notice that  $D$  is just a constant, thus can be removed without changing its performance. We call this effective ( $\max\text{SKSD-rg}$ ). As it only considers the marginal of the most important dimension, its variance does not increase with dimension of the problem. From the  $\max\text{SKSD-rg}$  value in figure.6 (left), we observe it has the lowest variance and does not change across the dimensions as expected. In terms of the rejection rate (the right of figure.6),  $\max\text{SKSD-rg}$  does not suffers from the curse-of-dimensionality, and consistently achieves nearly optimal rejection rate.

We also conduct the sensitivity test for  $\max\text{SKSD-rg}$  (the right in figure.5). We observe  $\max\text{SKSD-rg}$  not only addresses the mean shift pathology of  $\max\text{SKSD-g}$ , it is also more sensitive compared to KSD. Additionally, it is even more sensitive to the variance change compared to  $\max\text{SKSD-g}$ . The reason is the same as the mean shift case.

### G.4 IS MAXSKSD-RG ALWAYS BETTER?

Based on the above analysis,  $\max\text{SKSD-rg}$  is superior compared to  $\max\text{SKSD-g}$  theoretically. Indeed, it is trivial that  $\max\text{SKSD-g}$  is a lower bound for  $\max\text{SKSD-rg}$  which indicates a weaker discriminative power. However, this theoretical advantage relies on the assumption of the optimality of  $\mathbf{r}$ , which is also the key gap between theory and application.

$\max\text{SKSD-rg}$  often gives superior performance compared to  $\max\text{SKSD-g}$  in terms of GOF test. This is because GOF test only focus on the difference between two distributions, i.e. it focuses on finding a direction that gives higher discrepancy value. But this is not the case for model learning especially when they are used as training objectives. Instead, model learning focuses on the fact that the model approximates the target distribution in every directions. In theory,  $\max\text{SKSD-rg}$  can still give good performance as at each training iteration, this objective tries to minimize the largest difference between two distributions. However, this is not true in practice. We suspect the reasons are two fold: (1) optimal  $\mathbf{r}$  can not be guaranteed; (2) if the true optimal direction is drastically changing between iterations, gradient-based optimization may takes long time to move away from the current  $\mathbf{r}$ . Therefore, instead of relying on one slicing direction that might be sub-optimal in practice, in  $\max\text{SKSD-g}$  slicing along the directions in the orthogonal basis  $O_r$  provides better coverage of the difference between  $p$  and  $q$ . Figure 7 shows the comparison between  $\max\text{SKSD-rg}$  and  $\max\text{SKSD-g}$  for training 200 dimensional ICA.

## H COMPUTATIONAL AND MEMORY COST

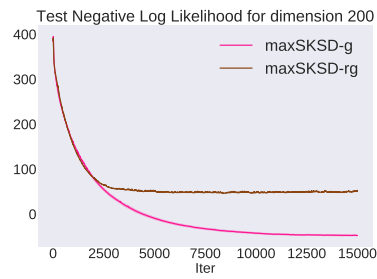


Figure 7: The ICA training curve of test negative LL with different training objectives. y-axis indicates the negative test log-likelihood.

In this section, we give a brief analysis on the computational and memory cost for performing GOF test and S-SVGD.

### H.1 MEMORY COST

First, let's consider the GOT test. If *maxSKSD-g* is used, we only need to store the slicing matrix  $\mathbf{G}$ , which cost  $O(D^2)$ . However, if *maxSKSD-rg* is used, the actual memory cost can be much reduced. For GOF test, we only care about whether it can detect the differences between distributions. Thus, we do not need  $D$  number of basis  $\mathbf{r}$ , instead, in theory, we only need 1 basis which maximizes the discriminative power. In practice, we may need  $m$  basis where  $1 < m \ll D$ . Thus, the memory cost will be  $O(mD) + O(mD) = O(mD)$ .

In the scope of this work, the memory cost for S-SVGD is similar to GOF test with *maxSKSD-g*, where slicing matrix  $\mathbf{G} \in \mathbb{R}^{D \times D}$  need to be stored. However, it remains a question that whether *maxSKSD-rg* can be used to derive a corresponding variant of S-SVGD. If that's the case, one only need to select 'important' basis  $\mathbf{r}$  to drive the particles towards the target distributions. Thus, the memory cost is reduced to  $O(mD)$ . For the BNN experiment in appendix J.5, the actual memory cost for SVGD with Boston housing data set is 1003MB and 1203MB for S-SVGD with NVIDIA Quadro P6000.

### H.2 TIME COMPLEXITY

The computational cost of computing the kernel  $K(\mathbf{x}, \mathbf{x}')$  with each pair of  $\mathbf{x}, \mathbf{x}'$  is  $O(D)$ . Thus, evaluating KSD requires the  $O(N^2D)$  where  $N$  is the number of samples used for Monte Carlo estimations. Similarly, the cost of each kernel evaluation  $K(\mathbf{x}^T \mathbf{g}, \mathbf{x}'^T \mathbf{g})$  is  $O(D)$ , but *maxSKSD-g* requires such evaluation for each  $\mathbf{g}$ . Thus, the total cost for *maxSKSD-g* is  $O(N^2D^2)$ . On the other hand, *maxSKSD-rg* can reduce such cost to  $O(N^2Dm)$  where  $m$  is the number of 'important' basis selected by maximizing Eq.11.

For SVGD, the kernel is also evaluated on each pair of  $\mathbf{x}, \mathbf{x}'$ . Thus, the computational cost is  $O(N^2D)$  where  $N$  is the number of particles. Similarly, the computational cost of S-SVGD is  $O(N^2D^2)$ . In practice, first we compare their time consumption on the above BNN experiment. The S-SVGD uses about 0.073s per epoch whereas SVGD only uses 0.032s per epoch, which is about 2.5 times larger than SVGD with P6000. For amortized SVGD experiment with MNIST data set, the time consumption of amortized SVGD is 0.112s per iteration, and S-SVGD is 0.122s per iteration, which is almost the same due to the smaller latent dimension (32).

We argue that there is no free lunch and every method has its compromises. In our case, the significant advantages of the proposed methods compared to KSD and SVGD come with the cost of higher computational and memory consumption. Even with this extra cost, the proposed framework is still a significant improvement for KSD and SVGD, as they fail even at very low dimensions (around 30), where the cost of our method is not much higher. Especially for GOF test, even the dimension is huge, one can always adopt *maxSKSD-rg* to reduce the memory and computational cost by selecting important basis.

## I GOF TEST

### I.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 (Jitkrittum et al., 2017; Huggins & Mackey, 2018). 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.(11)). 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 Multivariate-t test and 500 trials for Gaussian diffusion test.

### I.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 FSSD 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 I.3). Finding relatively good directions for maxSKSD with fewer training samples is a good direction for future work.

### I.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.

**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 8 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,

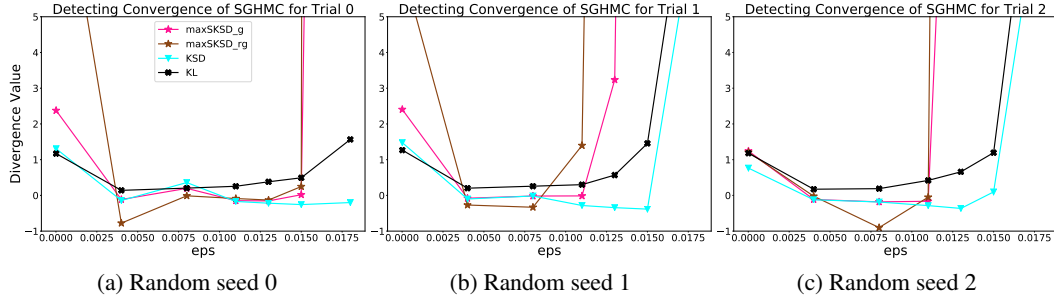


Figure 8: Discrepancy value for different random seed. The x-axis indicates the step size used for SGHMC.

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

(a)

Method	Metric				
	step size	KSD	maxSKSD_rg	maxSKSD_g	KL
KSD	0.015	-0.384	902	66.0	1.46
maxSKSD_rg	0.008	-0.0166	-0.332	-0.018	0.257
maxSKSD_g	0.004	-0.0100	-0.269	-0.079	0.201

(b)

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

## J MODEL TRAINING

### J.1 VARIANCE ESTIMATION FOR GAUSSIAN TOY EXAMPLE

This experiment is to demonstrate the mode collapse problem of SVGD at high dimensions and the advantage of the proposed S-SVG.

**Setup** The target distribution is an standard Gaussian distribution  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ . 50, 100 and 200 samples are used for SVGD and S-SVG. For fair comparison, we use the same RBF kernel with median heuristic for both SVGD and S-SVG. We run 6000 update steps to make sure they fully converged before estimating the variance. For S-SVG, to avoid the over-fitting of sliced matrix  $\mathbf{G}$  to small number of samples, we only update the matrix  $\mathbf{G}$  when the samples after the update are far away from the one used for previous update. The initialized particles are drawn from  $\mathcal{N}(\mathbf{2}, \mathbf{2I})$ . The evaluation metric is the averaged estimated variance of the resulting samples. Namely, for a  $D$  dimensional target distribution,  $\text{Var}_{avg} = \frac{1}{D} \sum_{d=1}^D \text{Var}(\{\mathbf{x}_d\}_{n=1}^N)$ .

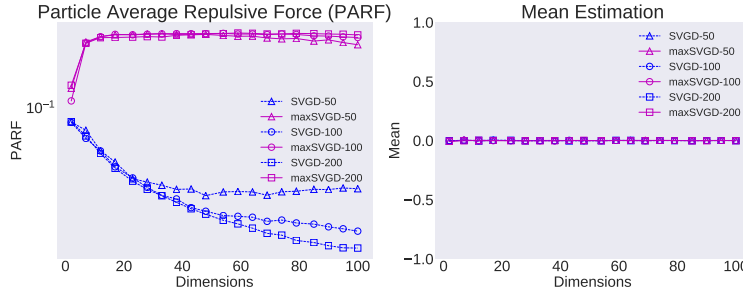


Figure 9: Statistics for SVGD and S-SVGd. The PARF is calculated as  $\frac{1}{N} \sum_{n=1}^N \|R(\mathbf{x}_n, \hat{q}_N)\|_\infty$  (Zhuo et al., 2017), where  $R(\mathbf{x}_n, \hat{q}_N)$  is the repulsive force for  $\mathbf{x}_n$  and  $\hat{q}_N$  is the empirical distribution of the samples  $\{\mathbf{x}\}_{n=1}^N$ . For SVGD, it is  $R(\mathbf{x}, q_y) = \mathbb{E}_{q_y}[\nabla_{\mathbf{y}} k(\mathbf{x}, \mathbf{y})]$  and for S-SVGd, the  $d^{\text{th}}$  element of  $R(\mathbf{x}, q_y)$  is  $R(\mathbf{x}, q_y)_d = \mathbb{E}_{q_y}[g_{d,d} \nabla_{\mathbf{y}^T} g_d^T k_{g_d}(\mathbf{x}^T \mathbf{g}_d, \mathbf{y}^T \mathbf{g}_d)]$ .

From figure 2, we observe when the sample number is small, the resulting samples tend to collapse to a point in high dimensions (low variance). On the other hand, the proposed S-SVGd correctly recovers the true target variance regardless of the number of samples and dimensions. This mode collapse behavior of SVGD is directly related to the decrease of the repulsive force at high dimensions (for detailed analysis of this behavior, refer to (Zhuo et al., 2017)). To verify this, we plot the *particle averaged repulsive force* (PARF) and the averaged estimated mean of the samples in figure 9. The PARF for SVGD reduces as the dimension increases whereas S-SVGd stays at a constant level. This is because the kernel and repulsive force of S-SVGd are evaluated on the one-dimensional projections instead of the full input  $\mathbf{x}$ . This dimensionality reduction side steps the decrease of the repulsive force at high dimensions regardless of the sample number, thus S-SVGd recovers the correct target variance.

## J.2 SETUP FOR ICA MODEL TRAINING

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 (Grathwohl et al., 2020) 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 (Grathwohl et al., 2020). 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 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.

## J.3 ICA ADDITIONAL PLOTS

From the figure 10, 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 with LSD and becomes faster after  $D = 60$ , whereas KSD starts to slow down and even diverges at  $D = 200$ .

## J.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

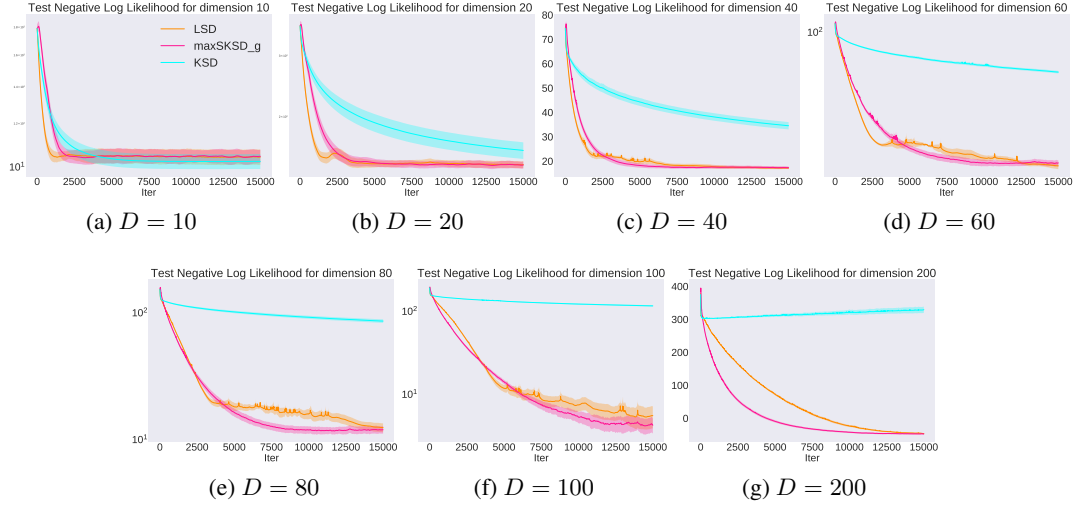


Figure 10: Training curve of different methods for ICA problems. The y-axis indicates the NLL of test data.

is chosen by the median heuristic. We update the sliced matrix  $\mathbf{G}$  for S-SVGd once per each encoder update. 50 latent samples are used for both encoder and decoder updates. We use Adam optimizer (Kingma & Ba, 2014) with 0.001 learning rate and 100 for batch size.

For evaluation, the log likelihood is computed using *Hamiltonian annealed importance sampling* (HAIS) (Wu et al., 2016). 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 (Rezende et al., 2014) to use approximate Gibbs sampler with  $D = 32$  latent space. Specifically, with missing and observed pixels denoted as  $\mathbf{x}_m$  and  $\mathbf{x}_o$ , encoder distribution  $q_\phi$  and decoder  $p_\theta$ , we iteratively applies the following procedure: (1) generate latent samples  $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}_o, \mathbf{x}_m)$  (2) reconstruction  $\mathbf{x}^* \sim p_\theta(\mathbf{x}^*|\mathbf{z})$  (3) Imputation  $\mathbf{x}_m \leftarrow \mathbf{x}_m^*$ . 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. Label entropy is computed by the 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  $\mathcal{L}$

**for**  $t \leq T$  **do**

Generate  $N$  initial latent samples using encoder  $\{\mathbf{z}_i\}_{i=1}^N = f_q(\mathbf{x})$ ;

Update the samples  $\{\mathbf{z}_i^*\}_{i=1}^N$  based on  $\{\mathbf{z}_i\}_{i=1}^N$  using SVGD or S-SVGd (algorithm 2) with step size  $\epsilon_S$ ;

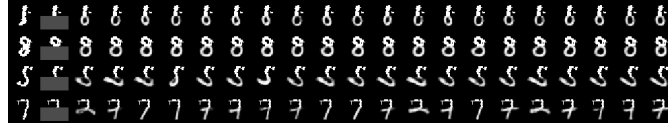
Compute the encoder MSE loss between  $\{\mathbf{z}_i^*\}_{i=1}^N$  and  $\{\mathbf{z}_i\}_{i=1}^N$  and update encoder  $f_q$  using Adam( $f_q, \epsilon_O$ );

Compute decoder loss  $\mathcal{L}(\mathbf{x}, \{\mathbf{z}_i^*\}_{i=1}^N)$  and update decoder using Adam( $f_d, \epsilon_O$ );

**end**

---

Figure 11 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



(a) Vanilla VAE



(b) Amortized SVGD



(c) Amortized S-SVGd

Figure 11: 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.

mode and get stuck at the wrong one (e.g. first digit '5'). This explains the slightly worse accuracy compared to amortized S-SVGd.

### J.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 (Liu & Wang, 2016) 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 for both SVGd and S-SVGd. For SVGd, the bandwidth is selected in the same way as (Liu & Wang, 2016). 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  $\sum_{1 \leq i < j \leq N} \text{dist}(\mathbf{x}_i, \mathbf{x}_j)$  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 datasets due to its better uncertainty estimation. For the remaining data set, e.g. *Combined*,

Table 5: BNN results on UCI regression benchmarks, comparing SVGD and S-SVGD. See main text for details.

Dataset	RMSE		test LL		Dist	
	SVGD	S-SVGD	SVGD	S-SVGD	SVGD	S-SVGD
Boston	$2.937 \pm 0.173$	<b><math>2.87 \pm 0.163</math></b>	$-2.533 \pm 0.092$	<b><math>-2.507 \pm 0.086</math></b>	$23272 \pm 986$	$49550 \pm 6250$
Concrete	$5.189 \pm 0.115$	<b><math>4.880 \pm 0.082</math></b>	$-3.076 \pm 0.024$	<b><math>-3.004 \pm 0.023</math></b>	$24650 \pm 1367$	$62680 \pm 1090$
Combined	$3.979 \pm 0.040$	<b><math>3.914 \pm 0.041</math></b>	$-2.802 \pm 0.010$	<b><math>-2.786 \pm 0.010</math></b>	$7148 \pm 245$	$33090 \pm 430$
Naval	$0.0030 \pm 0$	<b><math>0.0029 \pm 0</math></b>	$4.368 \pm 0.014$	<b><math>4.411 \pm 0.010</math></b>	$61838 \pm 2450$	$231600 \pm 2980$
Wine	$0.607 \pm 0.009$	<b><math>0.603 \pm 0.009</math></b>	$-0.924 \pm 0.015$	<b><math>-0.914 \pm 0.015</math></b>	$12534 \pm 982$	$35280 \pm 2470$
Energy	$1.353 \pm 0.049$	<b><math>1.132 \pm 0.048</math></b>	$-1.736 \pm 0.040$	<b><math>-1.540 \pm 0.044</math></b>	$16476 \pm 719$	$50850 \pm 1570$
kin8nm	$0.082 \pm 0.001$	<b><math>0.079 \pm 0</math></b>	$1.084 \pm 0.012$	<b><math>1.104 \pm 0.006</math></b>	$55715 \pm 2276$	$117700 \pm 902$
Yacht	$0.714 \pm 0.078$	<b><math>0.613 \pm 0.064</math></b>	$-1.277 \pm 0.155$	<b><math>-0.999 \pm 0.087</math></b>	$15530 \pm 1079$	$47290 \pm 2100$
Protein	<b><math>4.543 \pm 0.010</math></b>	$4.587 \pm 0.009$	<b><math>-2.932 \pm 0.003</math></b>	$-2.942 \pm 0.002$	$62370 \pm 2143$	$102600 \pm 2335$

*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.