Sobolev Space Regularised Pre Density Models

Mark Kozdoba [*1] **Binyamin Perets** [*1] **Shie Mannor** [1 2]

Abstract

We propose a new approach to non-parametric density estimation that is based on regularizing a Sobolev norm of the density. This method is statistically consistent, and makes the inductive bias of the model clear and interpretable. While there is no closed analytic form for the associated kernel, we show that one can approximate it using sampling. The optimization problem needed to determine the density is non-convex, and standard gradient methods do not perform well. However, we show that with an appropriate initialization and using natural gradients, one can obtain well performing solutions. Finally, while the approach provides pre-densities (i.e. not necessarily integrating to 1), which prevents the use of log-likelihood for cross validation, we show that one can instead adapt Fisher divergence based score matching methods for this task. We evaluate the resulting method on the comprehensive recent anomaly detection benchmark suite, ADBench, and find that it ranks second best, among more than 15 algorithms.

1. Introduction

Density estimation is one of the central problems in statistical learning. In recent years, there has been a tremendous amount of work in the development of parametric neural network based density estimation methods, such as Normalizing Flows (Papamakarios et al., 2021), Neural ODEs (Chen et al., 2018), and Score Based methods, (Song et al., 2021). However, the situation appears to be different for non parametric density estimation methods, (Wasserman, 2006), (Härdle et al., 2004). While there is recent work for low dimensional (one or two dimensional) data, see for instance (Takada, 2008), (Uppal et al., 2019), (Cui et al., 2020), (Marteau-Ferey et al., 2020), (Ferraccioli et al., 2021) (see also the survey (Kirkby et al., 2023)), there still are very few non-parametric methods applicable in higher dimensions.

Compared to parametric models, non parametric methods are often conceptually simpler, and the model bias (e.g., prior knowledge, type of smoothness) is explicit. This may allow better interpretability and better regularization control in small and medium sized data regimes.

Ideally, a function g produced by a density estimator should satisfy the following properties: *(i)* It is non-negative, *(ii)* it is integrable, and integrates to 1, *(iii)* it models well multidimensional data, and *(iv)* it is computationally feasible.

As mentioned above, to the best of our knowledge currently there are no non-parametric estimators that satisfy *(iii)* and *(iv)*. Moreover, note that even the requirements *(i)* and *(ii)* are non-trivial in the sense that they are not natural in the context of typical non-parametric constructions. Indeed, some of the above mentioned estimators simply forego either or both of these conditions even in 1d situations (see also a detailed discussion in (Marteau-Ferey et al., 2020) on this topic).

In this paper we develop the first non-parametric estimator that satisfies all the properties above, with the exception of integrating to 1 (but still integrable). Since many tasks derived from density estimation do not require the precise normalisation constant knowledge, this yields a practically useful and theoretically well founded method.

Let $\mathcal{S} = \{x_i\}_{i=1}^N \subset \mathbb{R}^d$ be a set of data points sampled i.i.d from some unknown distribution. We study a density estimator of the following form:

$$f^* := \operatorname*{argmin}_{f \in \mathcal{H}^a} -\frac{1}{N}\sum_{i=1}^{N} \log f^2(x_i) + \|f\|^2_{\mathcal{H}^a} . \quad (1)$$

Here $\mathcal{H}^a$ is a Sobolev type Reproducing Kernel Hilbert Space (RKHS) of functions, having a norm of the form

$$\|f\|^2_{\mathcal{H}^a} = \int_{\mathbb{R}^d} f^2(x)dx + a\int_{\mathbb{R}^d} |(Df)|^2(x)dx, \quad (2)$$

where D represents a combination of derivatives of a certain order. The density estimate is given by the function $(f^*)^2$. Note that $(f^*)^2$ is clearly non-negative, and $\|f\|_{\mathcal{H}^a} < \infty$

[*]Equal contribution [1]Technion Israel Institute of Technology, Haifa, Israel [2]NVIDIA Research. Correspondence to: Mark Kozdoba <mark.kozdoba@gmail.com>, Binyamin Perets <sbp67250@campus.technion.ac.il>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

implies $\int_{\mathbb{R}^d}(f^*)^2(x)dx < \infty$. Thus, $(f^*)^2$ is *integrable* over $\mathbb{R}^d$, although not necessarily integrates to 1. In this paper we refer to such functions as *pre-densities*. To convert such functions to densities, it sufficient to multiply them by the appropriate constant. Note also that (1) is essentially a regularized maximum likelihood estimate, where in addition to bounding the total mass of $(f^*)^2$, we also bound the norm of some of the derivatives of f^*. The fact that $\mathcal{H}^a$ is an RKHS allows us to compute f^* via the standard Representer Theorem. Observe that it would not be possible to control only the L_2 norm of f^* and maintain computabilty, since L_2 is not an RKHS. However, adding the derivatives with any coefficient $a > 0$ makes the space into an RKHS which allows to control smoothness, hence, we call the objective SObolev Space REgularised Pre density estimator (SOSREP).

The objective (1) has been introduced in (Good and Gaskins, 1971) and further studied in (Klonias, 1984), in the context of spline based methods in *one* dimension. In this paper we generalise this approach to arbitrary dimensions, which requires resolving several theoretical and computational issues.

On the side of the theory, generalising the existing 1d results, we prove the asymptotic consistency of the SOSREP estimator for the SDO kernels in any fixed dimension d, under mild assumptions on the ground truth density generating the x_i's. In addition, we present a family of examples in which the SOSREP and the standard Kernel Density Estimator (KDE, (Wasserman, 2006), with the same kernel) provably arbitrarily differ. Thus, SOSREP is a genuinely new estimator, non equivalent to KDE, and having some different properties. We also show that examples as above occur naturally in real datasets. Due to space constraints, the discussion of these examples is deferred to the Supplementary Material, Section F.

On the computation side, we resolve several issues: First, for $d > 1$ the kernel corresponding to $\mathcal{H}^a$, which we call the SDO kernel (Single Derivative Order; see Section 4), no longer has an analytical expression. We show however that it can, nevertheless, be approximated by an appropriate sampling procedure.

Next, the problem (1) is not convex in f, and we find that standard gradient descent optimization applied naively produces poor results. We show that this may be resolved by an appropriate initialization, and further improved by using a certain *natural gradient* optimisation rather than the standard one. Specifically, we will show that in our setup the natural gradient preserves some positivity properties, which are crucial for finding good solutions. Curiously, we believe this is one of the very few instances where it is clear *why* natural gradients perform better.

Further, note that as a result of consistency, it can also be shown that the estimator $(f^*)^2$ becomes normalized, at least asymptotically with N (see Section 5). However, in general computing or even estimating the normalization constant is not straightforward in terms of numerical computation, and is outside the scope of this paper [1]. Instead, we will focus on the Anomaly Detection (AD) applications, which do not require a normalization. A discussion of possible applications of SOSREP to another key task, that of generative modelling, is given in Section 7.

Next, although the normalisation may not be required for the main task itself, lack of normalisation may still introduces a particular nuisance in the context of hyperparameter tuning, as it prevents the utilization of the maximum likelihood measure to establish the optimal “bandwidth" parameter a. To resolve this, instead of the likelihood we consider the Fisher Divergence (FD), which uses log-likelihood *gradients* for divergence measurement, thereby eliminating the need for normalization. More specifically, we adapt the concept of score-matching (Hyvärinen, 2005; Song et al., 2020; Song and Ermon, 2019), a technique that has recently garnered renewed interest, to our setting.

Finally, building on the above steps, we show that SOSREP achieves the remarkable performance of scoring *second best* on a recent comprehensive anomaly detection benchmark, (Han et al., 2022) for tabular data, which includes 47 datasets and 15 specialized AD methods.

Interestingly, it is worth noting that anomaly detection has been regarded in the literature as a particularly difficult task, especially for deep density estimators (Nalisnick et al., 2019; Choi et al., 2019), and the difficulty persists even in the presence of substantial amounts data. We believe that this makes the good performance of a much more interpretable method, such as SOSREP, even more remarkable.

The rest of the paper is organized as follows: Section 2 reviews related literature. Section 3 introduces the RSR estimator and treats associated optimization questions. The SDO kernel and the associated sampling approximation are discussed in Section 4. Consistency results are stated in Section 5. Section 6 contains the experimental results, and Section 7 concludes the paper.

2. Literature and Related Work

As discussed in Section 1, a scheme that is equivalent to (1) was studied in (Good and Gaskins, 1971) and (Klonias, 1984); see also (Eggermont et al., 2001). However, these works concentrated solely on 1d case, and used spline methods to solve (3) in the special case that amounts to the use

[1] However, see Section 3.4, where an approach to an explicit normalisation at fixed finite N is discussed, along with other work were similar normalisation considerations are employed

of one particular kernel. More recently, an estimator that is closely related to (1) was considered in (Ferraccioli et al., 2021), but was restricted to the 2 dimensional case, both theoretically and practically. In particular, to obtain solutions, their approach involves discretizing (triangulating) the domain of interest in $\mathbb{R}^2$. This would not be feasible in any higher dimension. In contrast to these approaches, our more general RKHS formulation in Section 3.1 allows the use of a variety of kernels, and our optimisation algorithm is suitable, and was evaluated on, high dimensional data.

The work that is perhaps the most closely related to ours is (Marteau-Ferey et al., 2020), where the authors consider quadratic functions and regularisation with a version of an RKHS norm. The differences with respect to our work are both in scope and in details of the construction that enable computation. First, from a computational viewpoint, the objective in (Marteau-Ferey et al., 2020) requires optimisation over a certain family of non-negative matrices. While considering matrices has some advantages in terms of convexity, for a dataset of size N this requires N^2 parameters to encode such a matrix (their Theorem 1), and $O(N^3)$ computational cost per step ((Marteau-Ferey et al., 2020) top of page 6). Further, their optimisation is a constrained one, which makes the problem significantly more difficult. It thus would be unpractical to apply their methods to something like the ADBench benchmark, and indeed, they have only experimented with a 1-dimensional Gaussian example with $N = 50$. In contrast, similarly to the standard kernel methods, our objective only requires N parameters, the optimisation step is $O(N^2)$, and the optimisation can be done with standard optimisers, such as GD or Adam. Correspondingly, our evaluation is done on a state-of-the-art outlier detection dataset. Second, while (Marteau-Ferey et al., 2020) is concerned with non negative functions generally, we focus on density estimation much more closely. In particular, we prove consistency of our estimator, which was not shown in (Marteau-Ferey et al., 2020) for the estimator introduced there. In addition, we prove, both theoretically and through empirical evaluation, that our method is different from KDE, which is the most common non parametric density estimator.

The recent work (Tsuchida et al., 2024) may be viewed as neural network based variant of the construction in (Marteau-Ferey et al., 2020). Specifically, with slight simplification, in this work densities of the form $\langle v, \sigma\left(Wt(x) + b\right)\rangle^2$ are considered, where $t : \mathbb{R}^d \to \mathbb{R}^D$ is a fixed feature extractor, $W \in \mathbb{R}^{n\times D}$ and $b \in \mathbb{R}^D$ are linear and bias terms, viewed as a shallow neural network, σ is a non-linearity, and $v \in \mathbb{R}^n$. It is observed that for several possible choices of t, σ, the normalisation constant of such a density may be analytically computed as a function of v, W, b, thus making it an interesting density model, optimisable with maximum likelihood. Note that $\sigma\left(Wt(x) + b\right)$ may be viewed as a (learnable, but shallow) feature embedding into $\mathbb{R}^n$, and v as the linear combination of features, thus highlighting the analogy with the methods of (Marteau-Ferey et al., 2020) and with the approach in this paper. However, note also that the layer output dimension n here is involved in a trade-off: For higher n, the density family is more expressive, but the computation of the normalisation also becomes more complex (see Section 3.4). We note that in (Tsuchida et al., 2024) this approach was only evaluated as a density estimator on low dimensional data, and thus, in particular in view of the trade-off above, its behavior in higher dimensions is not clear at the moment.

The paper (Sriperumbudur et al., 2017) studies density estimation in the *extended exponential family*. That is, the densities are modeled as $q_0(x)e^{f(x)}$, where q_0 is a fixed known reference density on $\mathbb{R}^d$, and f varies in an RKHS $\mathcal{H}$. In (Sriperumbudur et al., 2017) it is argued that maximum-likelihood estimators are impractical due to the need to compute the normalisation, and therefore a Fisher Divergence (i.e. score-based) objective is used for the estimation. In contrast, surprisingly, in this work we show that in fact one *can* use a likelihood-like objective (1) to obtain a useful consistent estimator. [2] We emphasize that both their and our estimators produce unnormalised densities. The computation of the estimator of (Sriperumbudur et al., 2017) involves a solution of a certain linear system, which, while conceptually simple problem, is a computationally difficult due to the dimensions of this system. A number of variants with better computational complexity were developed in follow up work, see (Sutherland et al., 2018), (Dai et al., 2019), and (Singh et al., 2018).

Another common group of non parametric density estimators are the *projection methods*, (Wainwright, 2019). These methods have mostly been studied in one dimensional setting, see the survey (Kirkby et al., 2023). It is worth noting that with the exception of (Uppal et al., 2019), the estimators produced by these methods are not densities, in the sense that they do not integrate to 1, but more importantly, may take negative values. In the context of minmax bounds, projection methods in high dimensions were recently analyzed in (Singh et al., 2018), extending a classical work (Kerkyacharian and Picard, 1993). However, to the best of our knowledge, such methods have never been practically applied in high dimensions.

Consistency of the SOSREP in one dimension was shown in (Klonias, 1984), for kernels that coincide with SDO in one dimension. Our results provide L_2 consistency of the SOSREP in any fixed dimension d. While our approach generally follows the same lines as that of (Klonias, 1984),

[2] Note that while we also use FD in the experiments, we only use it to tune a single scalar parameter, which is computationally a considerably simpler problem than learning the whole density.

some of the estimates are done differently, since the corresponding arguments in (Klonias, 1984) were intrinsically one dimensional.

3. The SOSREP Desnity Estimator

In this Section we describe the general SOSREP Density Estimation Framework, formulated in an abstract Reproducing Kernel Hilbert Space. We then introduce and resolve issues related to the optimisation of the SOSREP objective, and discuss connections between convexity, positivity, and natural gradients.

3.1. The Basic Framework

Let $\mathcal{X}$ be a set and let $\mathcal{H}$ be a Reproducing Kernel Hilbert Space (RKHS) of functions on $\mathcal{X}$, with kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. In particular, $\mathcal{H}$ is equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and for every $x \in \mathcal{X}$, the function $k(x, \cdot) = k_x(\cdot) : \mathcal{X} \to \mathbb{R}$ is in $\mathcal{H}$ and satisfies the reproducing property, $\langle k_x, f \rangle_{\mathcal{H}} = f(x)$ for all $f \in \mathcal{H}$. The norm on $\mathcal{H}$ is denoted by $\|f\|^2_{\mathcal{H}} = \langle f, f \rangle_{\mathcal{H}}$, and the subscript $\mathcal{H}$ may be dropped when it is clear from context. We refer to (Schölkopf et al., 2002) for a general introduction to RKHS theory.

Given a set of points $S = \{x_1, \dots, x_N\} \subset \mathcal{X}$, we define the SOSREP estimator as the solution to the following optimization problem:

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} -\frac{1}{N} \sum_i \log f^2(x_i) + \|f\|^2_{\mathcal{H}} \,. \tag{3}$$

As discussed in Section 1, for appropriate spaces $\mathcal{H}$, the function $(f^*)^2$ corresponds to a pre-density (that is, $\int_{\mathbb{R}^d} (f^*)^2(x)dx < \infty$, but not necessarily $\int_{\mathbb{R}^d} (f^*)^2(x)dx = 1$). We now discuss a few basic properties of the solution to (3). First, by the Representer Theorem for RKHS, the minimizer of (3) has the form

$$f(x) = f_\alpha(x) = \sum_{i=1}^N \alpha_i k_{x_i}(x) \tag{4}$$

for some $\alpha = (\alpha_1, \dots, \alpha_N) \in \mathbb{R}^N$. Thus one can solve (3) by optimizing over a finite dimensional vector α. Next, it is worth noting that standard RKHS problems, such as regression, typically use the term $\lambda \|h\|^2_{\mathcal{H}}$, where $\lambda > 0$ controls the regularization strength. However, due to the special structure of (3), any solution with $\lambda \neq 1$ is a rescaling by a constant of a $\lambda = 1$ solution. Thus considering only $\lambda = 1$ in (3) is sufficient. In addition, we note that any solution of (3) satisfies $\|f\|^2_{\mathcal{H}} = 1$. See Lemma 4 in Supplementary Material for full details on these two points.

3.2. Convexity, Positivity, and Natural Gradients

Next, observe that the objective

$$\begin{aligned} L(f) &= -\frac{1}{N} \sum_i \log f^2(x_i) + \|f\|^2_{\mathcal{H}} \\ &= -\frac{1}{N} \sum_i \log \langle f, k_{x_i} \rangle^2_{\mathcal{H}} + \|f\|^2_{\mathcal{H}} \end{aligned} \tag{5}$$

is not convex in f. This is due to the fact that the scalar function $a \mapsto -\log a^2$ from $\mathbb{R}$ to $\mathbb{R}$ is not convex and is undefined at 0. However, the restriction of $a \mapsto -\log a^2$ to $a \in (0, \infty)$ is convex. Similarly, the restriction of L to the positive cone of functions $\mathcal{C} = \{f \mid f(x) \geq 0 \ \ \forall x \in \mathcal{X}\}$ is convex. Empirically, we have found that the lack of convexity results in poor solutions found by gradient descent. Intuitively, this is caused by f changing sign, which implies that f should pass through zero at some points. If these points happen to be near the test set, this results in low likelihoods. At the same time, there seems to be no computationally affordable way to constrain the optimization to the positive cone $\mathcal{C}$. Indeed, standard methods for constrained problems, such as the projected gradient descent, (Nesterov, 2003), would be costly due to the need to compute the projections.

We resolve this issue in two steps: First, we use a non-negative α initialization, $\alpha_i \geq 0$. Note that for f given by (4), if the kernel is non-negative, then f is non-negative. Although some kernels are non-negative, the SDO kernel, and especially its finite sample approximation (Section 4.2) may have negative values. At the same time, there are few such values, and empirically such initialization tremendously improves the performance of the gradient descent. Second, we use the *natural gradient*, defined in the next section. One can show that for non-negative kernels, $\mathcal{C}$ is in fact invariant under natural gradient steps (supplementary material Section H). That is, if for f_α given by (4) we have $f_\alpha \in \mathcal{C}$, and α' is obtained by the natural gradient step (8), then $f_{\alpha'} \in \mathcal{C}$. This does not seem to be true for the regular gradient. As a consequence, at least for fully non-negative kernels, by using natural gradient we can perform optimisation in $\mathcal{C}$, *without the need for computationally costly constraints*. As for the SDO kerenl, although it is not completely non-negative, we observe empirically that the use of natural gradient results in a more stable algorithm and in performance improvement compared to the standard gradient descent. A comparison of standard and natural gradients w.r.t stability to negative values is given in Section 6.3.

3.3. Explicit Form of Gradients

In this Section we explicitly compute the regular and the natural gradients, both in the function space and in terms of α.

We are interested in the minimization of $L(f)$, defined by (5). Using the representation (4) for $x \in \mathcal{X}$, we can equivalently consider minimization in $\alpha \in \mathbb{R}^N$. Let $K = \{k(x_i, x_j)\}_{i,j \leq N} \in \mathbb{R}^{N \times N}$ denote the empirical kernel matrix. Then standard computations show that $\|f_\alpha\|^2_{\mathcal{H}} = \langle K\alpha, \alpha \rangle_{\mathbb{R}^N}$ and we have $(f_\alpha(x_1), \ldots, f_\alpha(x_N)) = K\alpha$ (as column vectors). Thus one can consider $L(f_\alpha) = L(\alpha)$ as a functional $L : \mathbb{R}^N \to \mathbb{R}$ and explicitly compute the gradient w.r.t α. This gradient is given in (6).

As detailed in 3.1, it is also useful to consider the Natural Gradient – the gradient of $L(f)$ as a function of f, directly in the space $\mathcal{H}$. Briefly, a directional Fréchet derivative, (Munkres, 2018), of L at point $f \in \mathcal{H}$ in direction $h \in \mathcal{H}$ is defined as the limit $D_h L(f) = \lim_{\varepsilon \to 0} \varepsilon^{-1} \cdot (L(f + \varepsilon h) - L(f))$. As a function of h, $D_h L(f)$ can be shown to be a bounded and linear functional, and thus by the Riesz Representation Theorem, there is a vector, which we denote $\nabla_f L$, such that $D_h L(f) = \langle \nabla_f L, h \rangle$ for all $h \in \mathcal{H}$. We call $\nabla_f L$ the Natural Gradient of L, since its uses the native space $\mathcal{H}$. Intuitively, this definition parallels the regular gradient definition, but uses the $\mathcal{H}$ inner product to define the vector $\nabla_f L$, instead of the standard, "parametrization dependent" inner product in $\mathbb{R}^N$, that is used to define $\nabla_\alpha L$. For the purposes of this paper, it is sufficient to note that similarly to the regular gradient, the natural gradient satisfies the chain rule, and we have $\nabla_f \|f\|^2_{\mathcal{H}} = 2f$ and $\nabla_f \langle g, f \rangle_{\mathcal{H}} = g$ for all $g \in \mathcal{H}$. The explicit gradient expressions are given below:

Lemma 1 (Gradients). The standard and the natural gradients of $L(f)$ are given by

$$\begin{aligned} \nabla_\alpha L &= 2\left[K\alpha - \frac{1}{N} K \left(K\alpha\right)^{-1}\right] \in \mathbb{R}^N \text{ and} \\ \nabla_f L &= 2\left[f - \frac{1}{N}\sum_{i=1}^{N} f^{-1}(x_i) k_{x_i}\right] \in \mathcal{H} \end{aligned} \tag{6}$$

where for a vector $v \in \mathbb{R}^d$, v^{-1} means coordinatewise inversion.

If one chooses the functions k_{x_i} as a basis for the space $H_S = span\{k_{x_i}\}_{i \leq N} \subset \mathcal{H}$, then α in (4) may be regarded as coefficients in this basis. For $f = f_\alpha \in H_S$ one can then write in this basis $\nabla_f L = 2\left[\alpha - \frac{1}{N}(K\alpha)^{-1}\right] \in \mathbb{R}^N$. Therefore in the α-basis we have the following standard and natural gradient iterations, respectively:

$$\alpha' \leftarrow \alpha - 2\lambda\left[K\alpha - \frac{1}{N} K \left(K\alpha\right)^{-1}\right] \text{ and} \tag{7}$$

$$\alpha' \leftarrow \alpha - 2\lambda\left[\alpha - \frac{1}{N}\left(K\alpha\right)^{-1}\right], \tag{8}$$

where λ is the learning rate.

3.4. Normalisation

Let us now discuss the normalisation of densities f^2 with f given by (4). Since the normalising constant for such a density is simply $\int_{\mathbb{R}^d} f^2(x)dx$, we can write

$$\int_{\mathbb{R}^d} f^2(x)dx = \sum_{i,j \leq N} \alpha_i \alpha_j \int_{\mathbb{R}^d} k_{x_i}(x) k_{x_j}(x) dx = \left\langle \alpha, \bar{K}\alpha \right\rangle,$$

where $\bar{K} \in \mathbb{R}^{N \times N}$ is the matrix with entries $\bar{K}_{ij} = \int k_{x_i}(x) k_{x_j}(x) dx$. For the Gaussian kernel, these entries may be computed analytically, but for other kernels the situation is less clear. Another significant issue is that the operation $\left\langle \alpha, \bar{K}\alpha \right\rangle$ involves summation and multiplication of $O(N^2)$ terms, and as such is likely to be numerically unstable for larger N. Due to these reasons, we believe that normalisation requires further study, and is out of scope for this work.

It is worth noting, however, that in (Marteau-Ferey et al., 2020) a similar argument was used to achieve normalisation, with a small N and Gaussian kernels. Moreover, in (Tsuchida et al., 2024) (see Section 2), the normalisation constant also has a form $\left\langle \alpha, \bar{K}\alpha \right\rangle$ for an appropriate α. In this case, however, the matrix $\bar{K}$ depends on the parameters W, b instead of the inputs, and is of dimension $n \times n$, where n is the output size of the single hidden layer. We note in particular that since every entry of $\bar{K}$ depends on the parameters, the underlying computation graph of the autodifferentiation engine would scale as n^2, thus explaining the expressivity-computational complexity trade-off of this method, as discussed in Section 2.

4. Single Derivative Order Kernel Approximation

In this Section we introduce the Single Derivative Order kernel, which corresponds to norms of the form (2) discussed in Section 1. In Section 4.1 we introduce the relevant Sobolev functional spaces and derive the Fourier transform of the norm. In Section 4.2 we describe a sampling procedure that can be used to approximate the SDO.

4.1. The Kernel in Integral Form

For a function $f : \mathbb{R}^d \to \mathbb{C}$ and a tuple $\kappa \in (\mathbb{N} \cup \{0\})^d$, let $D^\kappa = \frac{\partial f}{\partial x_1^{\kappa_1} \ldots \partial x_d^{\kappa_d}}$ denote the κ indexed derivative. By convention, for $\kappa = (0, 0, \ldots, 0)$ we set $D^\kappa f = f$. Set also $\kappa! = \prod_{j=1}^d \kappa_j!$ and $|\kappa|_1 = \sum_{j=1}^d \kappa_j$. Set $\|f\|^2_{L_2} = \int |f(x)|^2 dx$. Then, for $m \in \mathbb{N}$ and $a > 0$ denote

$$\|f\|^2_a = \|f\|^2_{L_2} + a \sum_{|\kappa|_1 = m} \frac{m!}{\kappa!} \|(D^\kappa f)\|^2_{L_2}. \tag{9}$$

The norm $\|f\|^2_a$ induces a topology that is equivalent to that of a standard L_2 Sobolev space of order m. We refer to

(Adams and Fournier, 2003), (Saitoh and Sawano, 2016) for background on Sobolev spaces. However, here we are interested in properties of the norm that are finer than the above equivalence. For instance, note that for all $a \neq 0$ the norms $\|f\|_a$ are mutually equivalent, but nevertheless, a specific value of a is crucial in applications, for regularization purposes.

Let $\mathcal{H}^a = \left\{ f : \mathbb{R}^d \to \mathbb{C} \mid \|f\|_a^2 < \infty \right\}$ be the space of functions with a finite $\|f\|_a^2$ norm. Denote by

$$\langle f, g \rangle_{\mathcal{H}^a} = \langle f, g \rangle_{L_2} + a \sum_{|\kappa|_1 = m} \frac{m!}{\kappa!} \langle (D^\kappa f), (D^\kappa g) \rangle_{L_2} \tag{10}$$

the inner product that induces the norm $\|f\|_a^2$.

Theorem 2. For $m > d/2$ and any $a > 0$, the space $\mathcal{H}^a$ admits a reproducing kernel $k^a(x, y)$ satisfying $\langle k_x^a, f \rangle_{\mathcal{H}^a} = f(x)$ for all $f \in \mathcal{H}^a$ and $x \in \mathbb{R}^d$. The kernel is given by

$$k^a(x, y) = \int_{\mathbb{R}^d} \frac{e^{2\pi i \langle y - x, z \rangle}}{1 + a \cdot (2\pi)^{2m} \|z\|^{2m}} dz = \int_{\mathbb{R}^d} \frac{1}{1 + a \cdot (2\pi)^{2m} \|z\|^{2m}} \cdot e^{2\pi i \langle y, z \rangle} \cdot \overline{e^{2\pi i \langle x, z \rangle}} dz. \tag{11}$$

The proof of Theorem 2 follows the standard approach of deriving kernels in Sobolev spaces, via computation and inversion of the Fourier transform, see (Saitoh and Sawano, 2016). However, compact expressions such as (11) are only possible for some choices of derivative coefficients. Since the particular form (9) was not previously considered in the literature (except for $d = 1$, see below), we provide the full proof in the Supplementary Material.

4.2. Kernel Evaluation via Sampling

To solve the optimization problem (3) in $\mathcal{H}^a$, we need to be able to evaluate the kernel k^a at various points. For $d = 1$, closed analytic expressions were obtained in cases $m = 1, 2, 3$ in (Thomas-Agnan, 1996). In particular, for $m = 1$, k^a coincides with the Laplacian kernel $k_h(x, y) = e^{-h|x-y|}$. However, for $d > 1$, it seems unlikely that there are closed expressions. See (Novak et al., 2018) for a discussion of this issue for a similar family of norms.

To resolve this, note that the form (11) may be interpreted as an average of the terms $e^{2\pi i \langle y, z \rangle} \cdot \overline{e^{2\pi i \langle x, z \rangle}}$, where z is sampled from an unnormalized density $w^a(z) = (1 + a \cdot (2\pi)^{2m} \|z\|^{2m})^{-1}$ on $\mathbb{R}^d$. This immediately suggest that if we can sample from $w^a(z)$, then we can approximate k^a by summing over a finite set of samples z_j instead of computing the full integral.

In fact, a similar scheme was famously previously employed in (Rahimi and Recht, 2007), in a different context. There, it was observed that by Bochners's Theorem, (Rudin, 2017), any stationary kernel can be represented as $k(x, y) = \int \nu(z) e^{2\pi i \langle y, z \rangle} \cdot \overline{e^{2\pi i \langle x, z \rangle}} dz$ for some non-negative measure ν. Thus, if one can sample $z_1, \ldots, z_T$ from ν, one can construct an approximation

$$\hat{k}^a(x, y) = \frac{1}{T} \sum_{t=1}^{T} \cos\left(\langle z_t, x \rangle + b_t\right) \cdot \cos\left(\langle z_t, y \rangle + b_t\right), \tag{12}$$

where b_t are additional i.i.d samples, sampled uniformly from $[0, 2\pi]$. In (Rahimi and Recht, 2007), this approximation was used as a dimension reduction for *known* analytic kernels, such as the Gaussian, for which the appropriate ν are known. Note that the samples z_t, b_t can be drawn once, and subsequently used for all x, y (at least in a bounded region, see the uniform approximation result in (Rahimi and Recht, 2007)).

For the case of interest in this paper, the SDO kernel, Bochner's representation is given by (11) in Theorem 2. Thus, to implement the sampling scheme (12) it remains to describe how one can sample from the density $w^a(z)$ on $\mathbb{R}^d$. To this end, note that $w^a(z)$ is spherically symmetric, and thus can be decomposed as $z = r\theta$, where θ is sampled uniformly from a unit sphere S^{d-1} and the radius r is sampled from a *one dimensional* density $u^a(r) = \frac{r^{d-1}}{1 + a(2\pi r)^{2m}}$ (see the Supplementary Material for full details on this change of variables). Next, note that sampling θ is easy. Indeed, let $g_1, \ldots, g_d$ be i.i.d standard Gaussians. Then $\theta \sim (g_1, \ldots, g_d) / \sqrt{\sum_i g_i^2}$. Thus the problem is reduced to sampling a one dimensional distribution with a single mode, with known (unnormalized) density. This can be efficiently achieved by methods such as Hamiltonian Monte Carlo (HMC). However, we found that in all cases a fine grained enough discretization of the line was sufficient.

4.3. Relations With Other Kernels

We now discuss the relation between SDO and the Laplacian, Gaussian and Matérn kernels.

The SDO kernel with parameter m is defined by the norm (9), which involves only derivatives of order m and we use $m = \lceil d/2 \rceil$ in the experiments. On the other hand, it is well known that the Gaussian kernel, $k(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$, corresponds to a Sobolev space with a norm that is analogous to (9), but involves a summation over *all* orders of derivatives (see (Williams and Rasmussen, 2006), Chapter 6), and thus contains only infinitely smooth functions. More recently, it has been shown that the Laplacian kernel, $k(x, y) = e^{-\frac{\|x-y\|}{2\sigma}}$, corresponds to a Sobolev space with a norm that involves all the derivatives up to order $m = \lceil d/2 \rceil$, (see (Rakhlin and Zhai, 2019), Proposition 7). This norm, however, explicitly includes lower order deriva-

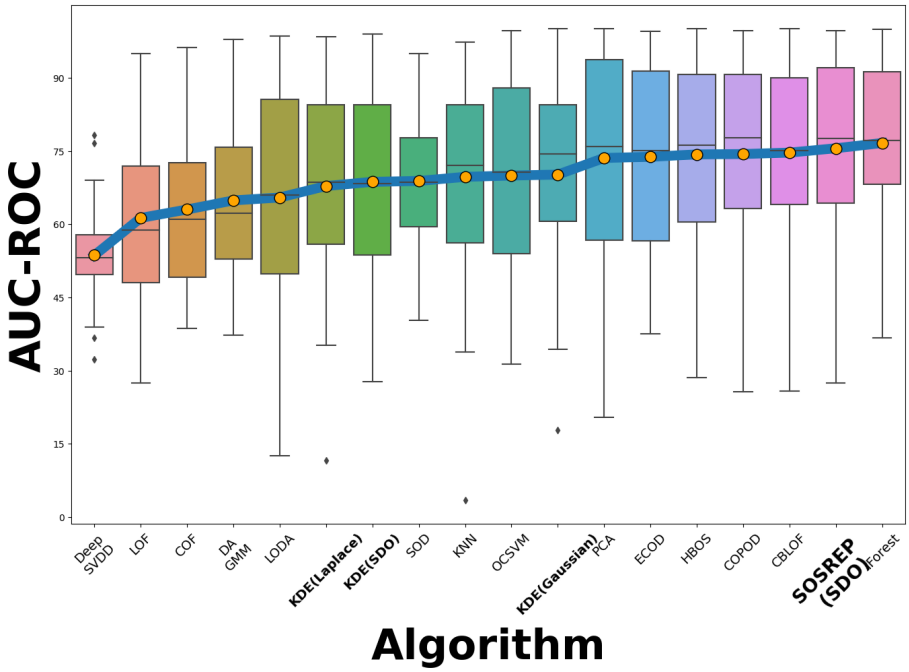

Figure 1: Anomaly Detection Results on ADBench, higher is Better. SOSREP is Second Best Among 18 Algorithms

tives ($m << d/2$) terms. Thus bounding the norm also explicitly bounds the lower derivative magnitudes. In this sense, the Laplace kernel penalises non-smoothness more heavily than the SDO. In fact, SDO with $m = \lceil d/2 \rceil$ may be viewed as the RKHS space with *minimal* smoothness requirements in d dimensions. Indeed, $m > d/2$ is necessary for the space to be an RKHS (otherwise (11) diverges, see Supplementary Sec. D.1 and D.2 for additional details), and no additional terms are present in the norm.

Finally, it is worth noting that the Matérn family of kernels with a smoothness parameter ν (Williams and Rasmussen, 2006) contains the Laplacian and Gaussian kernels as two extremes, with $\nu = \frac{1}{2}$ and $\nu \to \infty$, respectively. As shown in (Fasshauer and Ye, 2011), in fact the corresponding RKHS spaces have increasingly growing smoothness order, from $\lceil d/2 \rceil$ to ∞, when $\nu = \frac{1}{2} + p$, and p varies in $[0, \infty)$.

In addition to the SDO kernel, we have evaluated the SOSREP with Gaussian and Laplace kernels on the Anomaly Detection task and full results can be found in the Supplementary Material N. While the Gaussain and Laplacian kernels generally perform competitively and, in most cases, SOSREP estimators are better than KDE estimators, the performance of SDO is better. The precise reasons for this are not clear at the moment, and we believe that a detailed study of the interaction between the specific kernel and SOSREP is of interest. However, it is out of the scope of the present paper and is left as a future work.

5. Consistency

In this Section we describe the consistency result for the SOSREP estimators with kernels of the form (11).

Recall from the discussion in Section 3.2 that the objective L, given by (5), is convex when restricted to the positive cone. Here we consider the following positive cone:

$$\mathcal{C}' = \mathcal{C}'_a = \left\{ f \in span\left\{k^a_{x_i}\right\}_{i=1}^N \;\middle|\; f(x_i) > 0 \;\; \forall i \leq N \right\}. \tag{13}$$

Compared to Section 3.2, belonging to $\mathcal{C}'$ requires only positivity on the data points x_i rather than on all $x \in \mathbb{R}^d$, i.e. belonging to $\mathcal{C}'$ is a weaker requirement. The convexity of L still holds, however, due to similar considerations. In addition, we restrict the cone to the span of x_i since the optimal solutions belong to that span, and our algorithms operate inside the span too.

Note also that $f \in \mathcal{H}^1$ if and only if $f \in \mathcal{H}^a$ for every $a > 0$. With these remarks in mind, we can state the consistency result:

Theorem 3. Let $x_1, \ldots, x_N$ be i.i.d samples from a compactly supported density v^2 on $\mathbb{R}^d$, such that $v \in \mathcal{H}^1$. Set $a = a(N) = 1/N$, and let $u_N = u(x_1, \ldots, x_N; a(N))$ be the minimizer of the objective (5) in the cone (13). Then $\|u_N - v\|_{L_2}$ converges to 0 in probability.

In words, when N grows, and the regularisation size $a(N)$ decays as $1/N$, the SOSREP estimators u_N converge to v in L_2.

Note that since $\|v\|_{L_2} = 1$ (as its a density), and since

$\|u_N - v\|_{L_2} \to 0$, it follows by the triangle inequality that $\|u_N\|_{L_2} \to 1$. That is, the estimator u_N becomes approximately normalized as N grows.

An overview and full details of the proof are provided in Supplementary Material Sections M.1 and M.2.

6. Experiments

In this section, we present the evaluation of SOSREP on the ADBench anomaly detection benchmark, and empirically test the advantage of natural gradient descent for maintaining a non-negative f. The evaluations of SOSREP with different kernels on a 2-dimensional synthetic dataset are presented in the Supplementary Material section J.1.

6.1. Anomaly Detection Results for ADbench

ADbench (Han et al., 2022) is a recent Anomaly Detection (AD) benchmark evaluating 15 state-of-the-art algorithms on 47 tabular datasets. Alongside these, we also evaluate standard KDE estimators with Gaussian and Laplace kernels. The performance of all these algorithms is compared to that of SOSREP with the SDO kernel.

We focus on the classical unsupervised setup, in which at train time there is no information regarding which samples are anomalies. On test, anomaly labels are revealed only for final algorithm evaluation. For all density-based approaches (KDE,SOSREP), we employ the negative of the density as the *anomaly score*. That is, low density values would be considered anomalies. In accordance with the ADbench paper, we evaluate each dataset by calculating the average AUC-ROC of that score across four runs, each with a different random seed. The seed determines the train-test split of the ADbench data, as well as the randomness of the algorithms.

In Figure 1, each algorithm is represented by a box plot where the average AUC-ROC across all datasets is indicated by an orange dot. The boundaries of the box represent the 25th and 75th quantiles, while the line inside the box denotes the median. The algorithms are sorted by their average AUC-ROC. Additional evaluations are available in the Supplementary Material, Section J.2. These include a comparison of SOSREP using different kernels (Gaussian, Laplace, and SDO) against other methods on a per-dataset basis—contrary to the per-method analysis provided here. Furthermore, evaluations that assess the *relative ranks* of the algorithms, instead of the mean AUC-ROC, can also be found in the same section. The results there are very similar to those in Fig. 1. As discussed in (Han et al., 2022), evaluation by rank may help mitigate potential evaluation biases arising from variations in dataset difficulty.

For both AUC-ROC and rank evaluations, **SOSREP emerges as the 2nd best AD method overall**. Notably, this is achieved with the generic version of our method, without any pre or post-processing specifically dedicated to AD. In contrast, many other methods are specifically tailored for AD and do include extensive pre and post-processing.

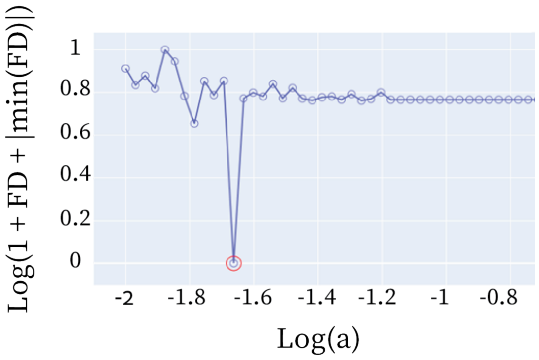

Figure 2: For each HP a, we calculate the Fisher divergence between the density learned on the training set and the density inferred on the test set.

In addition to performing well on the standard ADBench benchmark, and perhaps even more impressively, SOSREP excels also on the more demanding setup of *duplicate anomalies*, which was also extensively discussed in (Han et al., 2022). Here, **SOSREP rises to the forefront as the top AD method** (with an average AUC-ROC of 71.6 for X5 duplicates - a lead of 4% over the closest contender). This scenario, which is analogous to numerous practical situations, is a focal point for ADbench's assessment of unsupervised AD methods due to its inherent difficulty, leading to substantial drops in performance for former leaders like Isolation Forest. Detailed explanations are available in the Supplementary Material Section J.3.

As for computational cost, the entire set of 47 datasets was processed in 186 minutes using a single 3090RTX GPU and one CPU, averaging about 4 minutes per dataset.

6.2. Fisher-Divergence Based Hyperparameter Tuning

As discussed in Section 1, dealing with unnormalized densities adds some complexity to hyperparameter tuning. Indeed, hyperparameter tuning typically involves the comparison of multiple models, for instance models with different smoothness control (aka "bandwidth") parameters $a > 0$. In probabilistic models, this would typically involve choosing the model which gives the highest likelihood to the data (note that the anomaly labels are not available at this stage). However, due to the lack of normalisation, the likelihoods are not computable. In this paper, instead of likelihoods we compute the Fisher Diveregences between the model and the data, as these do not require normalisation, and choose the model with the parameter that yields the lowest diver-

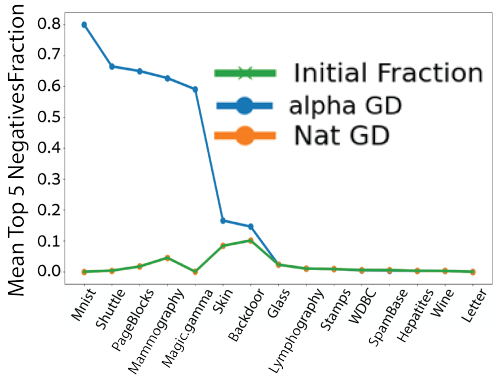

Figure 3: Fraction of negative values for natural versus α gradient-based optimization across datasets. The X-axis represents datasets from ADbench (see Supplementary Material Section L for details).

gence (more precisely, we compute the FD up to an additive constant which does not depend on the the model). Figure 2 illustrates the FD for SOSREP for different a values for a single dataset. The x-axis shows the parameter values on a logarithmic scale. For visual clarity, we also shifted our version of FD to be strictly positive and applied a logarithmic transformation. One can see that the graph is somewhat noisy, but typically there is one clear minimum value.

A detailed discussion of the stages involved in the FD computation and the associated minimum selection can be found in Section I of the supplementary material.

6.3. Natural-Gradient vs Standard Gradient Comparison

In the context of the discussion in Section 3.1, we conduct an experiment to demonstrate that the standard gradient descent may significantly amplify the fraction of negative values in a SOSREP solution, while the natural gradient keeps it constant. We have randomly chosen 15 datasets from ADBench, and for each dataset we have used 50 non negative α initializations. Then we ran both algorithms for 1000 iterations. The fraction of negative values of f_α (on the train set) was measured at initialization, and in the end of each run. In Figure 3, for each dataset and for each method, we show an average of the worst 5 fractions among the 50 initializations. Thus, for instance, for the 'shuttle' data, the initial fraction is negligible, and is unchanged by the natural gradient. However, the standard gradient ("alpha GD" in the figure in blue) for this dataset yields about 70% negative values in the 5 worst cases (i.e., 10% of initializations). We conclude that the natural gradient is better at preserving non-negativity than standard gradient descent.

7. Conclusion

In this paper we have developed the theory and the optimisation methods for a multi dimensional unnormalised density estimator. This is the first estimator, among those broadly based on variants of functional space normalisation, that is computationally tractable, can be applied in higher dimensions, and performs well on a useful and difficult task.

In particular, we have shown that the estimator is asymptotically consistent, that it is possible to use sampling to approximate the associated kernel, that it is possible to use a version of natural gradient for optimisation, and Fisher divergence for hyperparameter selection. We have found that the combination of these techniques result in a model that has excellent performance on a state-of-the-art outlier detection benchmark.

The success of the approach on the ADBench benchmarks indicates that it is capable of modelling complicated densities. It is thus natural to ask next whether the unnormalised densities introduced in this paper can be used as a basis for generative models. Generative modelling for tabular data of small and intermediate sizes is an active and difficult research area, and SOSREP estimators may provide an alternative approach to the problem. Indeed, note that most Markov Chain Monte Carlo methods (Metropolis-Hastings, Langevin and Hamiltonian MC, etc.) work with unnormalised densities by design and can in principle be applied to SOSREP estimated densities out of the box to generate samples. We believe that evaluating such approaches, and designing schemes to alleviate the "local minima" issues common to MCMC, is an important and promising direction for future research that is based on the present work.

All code necessary to replicate the results presented in this paper, as well as the implementation needed to run SOSREP with the kernels described herein, is publicly available. You can access and download the code from our GitHub repository at (https://github.com/bp6725/SOSREP). We hope this will enable others in the community to verify and build upon our work.

Acknowledgements

We would like to thank the anonymous reviewers for pointing out several relevant references, and for a number of other constructive suggestions that helped improve the quality and readability of this work.

This work was funded by the European Union under Horizon Europe grant number 101084642 and supported by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 101084642].

Impact Statement

This paper presents work which has the goal of advancing the fields of Density Estimation and Outlier Detection, and studies a general method, applicable across a variety of domains. While there may be many potential societal consequences related to the applications of the fields of study as above, we believe none of them are specific to our work, and thus do not need to be specifically highlighted here.

References

Adams, R. A. and Fournier, J. J. (2003). *Sobolev spaces*. Elsevier.

Betancourt, M. (2017). A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*.

Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., Assent, I., and Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Min. Knowl. Discov.*, 30(4):891–927.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Choi, H., Jang, E., and Alemi, A. A. (2019). Waic, but why? generative ensembles for robust anomaly detection.

Cui, Z., Kirkby, J. L., and Nguyen, D. (2020). Nonparametric density estimation by b-spline duality. *Econometric Theory*, 36(2):250–291.

Dai, B., Dai, H., Gretton, A., Song, L., Schuurmans, D., and He, N. (2019). Kernel exponential family estimation via doubly dual embedding. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2321–2330. PMLR.

Eggermont, P. P. B., LaRiccia, V. N., and LaRiccia, V. (2001). *Maximum penalized likelihood estimation. Volume I: Density Estimation*, volume 1. Springer.

Fasshauer, G. E. and Ye, Q. (2011). Reproducing kernels of generalized sobolev spaces via a green function approach with distributional operators. *Numerische Mathematik*, 119:585–611.

Ferraccioli, F., Arnone, E., Finos, L., Ramsay, J. O., Sangalli, L. M., et al. (2021). Nonparametric density estimation over complicated domains. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY SERIES B STATISTICAL METHODOLOGY*, 83(2):346–368.

Good, I. and Gaskins, R. A. (1971). Nonparametric roughness penalties for probability densities. *Biometrika*, 58(2):255–277.

Grafakos, L. (2008). *Classical fourier analysis*, volume 2. Springer.

Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. (2018). Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*.

Guha, S., Mishra, N., Roy, G., and Schrijvers, O. (2016). Robust random cut forest based anomaly detection on streams. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2712–2721. JMLR.org.

Han, S., Hu, X., Huang, H., Jiang, M., and Zhao, Y. (2022). Adbench: Anomaly detection benchmark.

Härdle, W., Müller, M., Sperlich, S., Werwatz, A., et al. (2004). *Nonparametric and semiparametric models*, volume 1. Springer.

Hutchinson, M. (1989). A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18(3):1059–1076.

Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).

Kerkyacharian, G. and Picard, D. (1993). Density estimation by kernel and wavelets methods: optimality of besov spaces. *Statistics & Probability Letters*, 18(4):327–336.

Kirkby, J. L., Leitao, Á., and Nguyen, D. (2023). Spline local basis methods for nonparametric density estimation. *Statistic Surveys*, 17:75–118.

Klonias, V. (1984). On a class of nonparametric density and regression estimators. *The Annals of Statistics*, pages 1263–1284.

Kwon, D., Natarajan, K., Suh, S. C., Kim, H., and Kim, J. (2018). An empirical study on network anomaly detection using convolutional neural networks. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1595–1598.

Loève, M. (1977). *Probability theory I*. Springer.

Lyu, S. (2012). Interpretation and generalization of score matching. *CoRR*, abs/1205.2629.

Marteau-Ferey, U., Bach, F., and Rudi, A. (2020). Nonparametric models for non-negative functions. *Advances in neural information processing systems*, 33:12816–12826.

Munkres, J. R. (2018). *Analysis on manifolds*. CRC Press.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. (2019). Do deep generative models know what they don't know?

Nesterov, Y. (2003). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Novak, E., Ullrich, M., Woźniakowski, H., and Zhang, S. (2018). Reproducing kernels of sobolev spaces on rd and applications to embedding constants and tractability. *Analysis and Applications*, 16(05):693–715.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.

Rakhlin, A. and Zhai, X. (2019). Consistency of interpolation with laplace kernels is a high-dimensional phenomenon. In *Conference on Learning Theory*, pages 2595–2623. PMLR.

Rudin, W. (2017). *Fourier analysis on groups*. Courier Dover Publications.

Saitoh, S. and Sawano, Y. (2016). *Theory of reproducing kernels and applications*. Springer.

Schölkopf, B., Smola, A. J., Bach, F., et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Singh, S., Uppal, A., Li, B., Li, C.-L., Zaheer, M., and Póczos, B. (2018). Nonparametric density estimation under adversarial losses. *Advances in Neural Information Processing Systems*, 31.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y., Garg, S., Shi, J., and Ermon, S. (2020). Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Sriperumbudur, B., Fukumizu, K., Gretton, A., Hyv, A., Kumar, R., et al. (2017). Density estimation in infinite dimensional exponential families. *Journal of Machine Learning Research*, 18(57):1–59.

Stein, E. M. and Weiss, G. (1971). Introduction to fourier analysis on euclidean spaces (pms-32), volume 32. In *Introduction to Fourier Analysis on Euclidean Spaces (PMS-32), Volume 32*. Princeton university press.

Sutherland, D. J., Strathmann, H., Arbel, M., and Gretton, A. (2018). Efficient and principled score estimation with nyström kernel exponential families. In *International Conference on Artificial Intelligence and Statistics*, pages 652–660. PMLR.

Takada, T. (2008). Asymptotic and qualitative performance of non-parametric density estimators: a comparative study. *The Econometrics Journal*, 11(3):573–592.

Thomas-Agnan, C. (1996). Computing a family of reproducing kernels for statistical applications. *Numerical Algorithms*, 13(1):21–32.

Tsuchida, R., Ong, C. S., and Sejdinovic, D. (2024). Squared neural families: A new class of tractable density models. *Advances in Neural Information Processing Systems*, 36.

Uppal, A., Singh, S., and Póczos, B. (2019). Nonparametric density estimation & convergence rates for gans under besov ipm losses. *Advances in neural information processing systems*, 32.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press.

Wasserman, L. (2006). *All of nonparametric statistics*. Springer Science & Business Media.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. MIT press Cambridge, MA.

A. Overview Of The Supplementary Material

This Supplementary Material is organized as follows:

- Some basic properties of the SOSREP objective and of the related minimization problem: Section B
- Derivation of the Gradients of the SOSREP objective: Section C
- Invariance of the non-negative function cone under natural gradient steps: Section H
- SOSREP vs KDE comparison: Sections F and G
- Derivation of the integral form of SDO kernel, proof of Theorem 3: Section D.1
- Additional details on sampling approximation procedure: Section D.2
- Details on the Fisher Divergence estimation for SOSREP Hyperparameter tuning : I
- Comparison of the raw AUC-ROC metric on ADBench data: Section J.2
- Discussion of an additional test regime, with duplicated anomalies: Section J.3
- On-the-fly hyperparameter tuning procedure that was used to save time by finding the first *stable* local minimum: Section J.6

The code used for all the experiments in the paper will be publicly released with the final version of the paper.

B. Basic Minimizer Properties

As discussed in Section 3.1, the minimizer of the SOSREP objective (3) always has $\mathcal{H}$ norm 1. In addition, there is no added value in multiplying the norm by a regularization scalar, since this only rescales the solution. Below we prove these statements.

Lemma 4. Define

$$f = \operatorname*{argmin}_{h \in \mathcal{H}} -\frac{1}{N} \sum_i \log h^2(x_i) + \|h\|_{\mathcal{H}}^2 . \tag{14}$$

Then f satisfies $\|f\|^2 = 1$. Moreover, if

$$f' = \operatorname*{argmin}_{h \in \mathcal{H}} -\frac{1}{N} \sum_i \log h^2(x_i) + \lambda^2 \|h\|_{\mathcal{H}}^2 , \tag{15}$$

for some $\lambda > 0$, then $f' = \lambda^{-1} f$.

Proof. For any $h \in \mathcal{H}$ and $a > 0$,

$$\operatorname*{argmin}_{a>0} -\frac{1}{N} \sum_i \log(ah)^2(x_i) + \|ah\|^2 = \tag{16}$$

$$\operatorname*{argmin}_{a} -\frac{1}{N} \sum_i \log h^2(x_i) - \log a^2 + a^2 \|h\|^2 . \tag{17}$$

Taking derivative w.r.t a we have

$$-\frac{2a}{a^2} + 2a \|h\|^2 = 0. \tag{18}$$

Thus optimal a for the problem (3) must satisfy $\|ah\|^2 = a^2 \|h\|^2 = 1$. To conclude the proof of the first claim, choose h in (16) to be the minimizer in (14), $h = f$. Note that if $\|f\|_{\mathcal{H}} \neq 1$, then we can choose $a = \|f\|_{\mathcal{H}}^{-1} \neq 1$ to further decrease the value of the objective contradicting the fact that f is the minimizer.

For the second claim, denoting $g = \lambda h$,

$$\underset{h\in\mathcal{H}}{\operatorname{argmin}} -\frac{1}{N}\sum_i \log h^2(x_i) + \lambda^2 \|h\|^2 \tag{19}$$

$$= \lambda^{-1} \underset{g\in\lambda\mathcal{H}=\mathcal{H}}{\operatorname{argmin}} -\frac{1}{N}\sum_i \log g^2(x_i) + \|g\|^2 + \frac{1}{N}\sum_i \log \lambda^2 \tag{20}$$

$$= \lambda^{-1} \underset{g\in\mathcal{H}}{\operatorname{argmin}} -\frac{1}{N}\sum_i \log g^2(x_i) + \|g\|^2 \tag{21}$$

$$= \lambda^{-1} f. \tag{22}$$

□

C. Derivation of the Gradients, Proof Of Lemma 1

In this section we derive the expressions for standard and the natural gradients of the objective (5), as given in Lemma 1.

Proof Of Lemma 1. We first derive the expression for $\nabla_\alpha L$ in (6). Recall that $\|f\|^2_{\mathcal{H}} = \langle \alpha, K\alpha\rangle_{\mathbb{R}^N}$ for $\alpha \in \mathbb{R}^N$, where $K_{ij} = k(x_i, x_j)$. This follows directly from the form (4), and the fact that $\langle k_x, k_y\rangle = k(x, y)$ for all $x, y \in \mathcal{H}$, by the reproducing property. For this term we have $\nabla_\alpha \langle \alpha, K\alpha\rangle = 2K\alpha$. Next, similarly by using (4), $\nabla_\alpha f(x) = (k(x_1, x), \ldots, k(x_N, x))$ for every $x \in \mathbb{R}^d$. Finally, we have

$$\nabla_\alpha \frac{1}{N}\sum_{i=1}^N \log f^2(x_i) = \frac{1}{N}\sum_{i=1}^N f^{-2}(x_i)\cdot 2f(x_i)\cdot \nabla_\alpha f(x_i) \tag{23}$$

$$= 2\frac{1}{N}\sum_{i=1}^N f^{-1}(x_i)\cdot \nabla_\alpha f(x_i) \tag{24}$$

$$= 2\frac{1}{N} K(f(x_1), \ldots, f(x_N))^{-1} \tag{25}$$

$$= 2\frac{1}{N} K\,(K\alpha)^{-1}. \tag{26}$$

This yields (6).

For $\nabla_f L$, we similarly have $\nabla_f \|f\|^2_{\mathcal{H}} = 2f$, as discussed in section 3.3. Moreover,

$$\nabla_f \frac{1}{N}\sum_{i=1}^N \log f^2(x_i) = \nabla_f \frac{1}{N}\sum_{i=1}^N \log \langle f, x_i\rangle^2_{\mathcal{H}} \tag{27}$$

$$= \frac{1}{N}\sum_{i=1}^N \langle f, x_i\rangle^{-2}_{\mathcal{H}} \cdot 2\langle f, x_i\rangle_{\mathcal{H}} \cdot \nabla_f \langle f, x_i\rangle_{\mathcal{H}} \tag{28}$$

$$= 2\frac{1}{N}\sum_{i=1}^N \langle f, x_i\rangle^{-1}_{\mathcal{H}}\, x_i. \tag{29}$$

This completes the proof. □

D. SDO Kernel Details

In Section D.1 we provide a full proof of Theorem 2, while Section D.2 contains additional details on the sampling approximation of SDO.

D.1. SDO Kernel Derivation

We will prove a claim that is slightly more general than Theorem 2. For a tuple $\bar{a} \in \mathbb{R}_+^m$, define the norm

$$\|f\|_{\bar{a}}^2 = \sum_{l=0}^{m} a_l \sum_{|\kappa|_1 = l} \frac{l!}{\kappa!} \|(D^\kappa f)\|_{L_2}^2 , \tag{30}$$

where D^κ are the κ-indexed derivative, as discussed in Section 4.1. The SDO norm is a special case with $a_0 = 1$, $a_m = a$, and $a_l = 0$ for $0 < l < m$. Let $\mathcal{H}^{\bar{a}}$ be the subspace of L_2 of functions with finite norm,

$$\mathcal{H}^{\bar{a}} = \{f \in L_2 \ \mid \ \|f\|_{\bar{a}} < \infty\} \tag{31}$$

and let the associated inner product be denoted by

$$\langle f, g \rangle_{\bar{a}} = \sum_{l=0}^{m} a_l \sum_{|\kappa|_1 = l} \frac{l!}{\kappa!} \langle (D^\kappa f), (D^\kappa g) \rangle_{L_2} . \tag{32}$$

Define the Fourier transform

$$\mathcal{F} f(z) = \int_{\mathbb{R}^d} f(u) e^{-2\pi i \langle z, u \rangle} du, \tag{33}$$

and recall that we have (see for instance (Stein and Weiss, 1971), (Grafakos, 2008))

$$\mathcal{F}\left(D^\kappa f\right)(z) = \left(\prod_{j=1}^{d} (2\pi i z_j)^{\kappa_j} \right) \mathcal{F} f(z) \text{ for all } z \in \mathbb{R}^d. \tag{34}$$

The following connection between the L_2 and the derivative derived norms is well known for the standard Sobolev spaces ((Williams and Rasmussen, 2006; Saitoh and Sawano, 2016; Novak et al., 2018)). However, since (30) somewhat differs from the standard definitions, we provide the argument for completeness.

Lemma 5. Set for $z \in \mathbb{R}^d$

$$v_{\bar{a}}(z) = \left(1 + \sum_{l=1}^{m} a_l \cdot (2\pi)^{2l} \|z\|^{2l} \right)^{\frac{1}{2}} . \tag{35}$$

Then for every $f \in \mathcal{H}^{\bar{a}}$ we have

$$\|f\|_{\bar{a}}^2 = \|v_{\bar{a}}(z) \cdot \mathcal{F}[f]\|_{L_2}^2 . \tag{36}$$

Proof.

$$\|f\|_{\bar{a}}^2 = \sum_{l=0}^{m} a_l \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \|D^\kappa f\|_{L_2}^2 \tag{37}$$

$$= \sum_{l=0}^{m} a_l \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \|\mathcal{F}[D^\kappa f]\|_{L_2}^2 \tag{38}$$

$$= \int dz \left[\sum_{l=0}^{m} a_l \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} |\mathcal{F}[D^\kappa f](z)|^2 \right] \tag{39}$$

$$= \int dz \left[|\mathcal{F}[f](z)|^2 + \sum_{l=1}^{m} a_l \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \left(\prod_{j=1}^{d} (2\pi z_j)^{2\kappa_j} \right) |\mathcal{F}[f](z)|^2 \right] \tag{40}$$

$$= \int dz\, |\mathcal{F}[f](z)|^2 \left[1 + \sum_{l=1}^{m} a_l \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \left(\prod_{j=1}^{d} (2\pi z_j)^{2\kappa_j} \right) \right] \tag{41}$$

$$= \int dz\, |\mathcal{F}[f](z)|^2 \left[1 + \sum_{l=1}^{m} a_l \cdot (2\pi)^{2l} \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \prod_{j=1}^{d} z_j^{2\kappa_j} \right] \tag{42}$$

$$= \int dz\, |\mathcal{F}[f](z)|^2 \left[1 + \sum_{l=1}^{m} a_l \cdot (2\pi)^{2l} \|z\|^{2l} \right] \tag{43}$$

□

Using the above Lemma, the derivation of the kerenl is standard. Suppose $k^{\bar{a}}$ is the kernel corresponding to $\|f\|_{\bar{a}}$ on $\mathcal{H}^{\bar{a}}$. It remains to observe that by the reproducing property and by Lemma 5, for all $x \in \mathbb{R}^d$

$$f(x) = \langle f, k_x^{\bar{a}} \rangle_{\bar{a}} \tag{44}$$

$$= \int_{\mathbb{R}^d} dz \;\; \mathcal{F}[f](z) \overline{\mathcal{F}[k_x^{\bar{a}}](z)} v_{\bar{a}}^2(z). \tag{45}$$

On the other hand, by the Fourier inversion formula, we have

$$f(x) = \int dz \;\; \mathcal{F}[f](z) e^{2\pi i \langle x, z \rangle}. \tag{46}$$

This implies that

$$\int dz \;\; \mathcal{F}[f](z) e^{2\pi i \langle x, z \rangle} = \int_{\mathbb{R}^d} dz \;\; \mathcal{F}[f](z) \overline{\mathcal{F}[k_x^{\bar{a}}](z)} v_{\bar{a}}^2(z) \tag{47}$$

holds for all $f \in \mathcal{H}^{\bar{a}}$, which by standard continuity considerations yields

$$\mathcal{F}[k_x^{\bar{a}}](z) = \frac{e^{-2\pi i \langle x, z \rangle}}{v_{\bar{a}}^2(z)}. \tag{48}$$

Using Fourier inversion again we obtain

$$k^{\bar{a}}(x, y) = \int_{\mathbb{R}^d} \frac{e^{2\pi i \langle y - x, z \rangle}}{v_{\bar{a}}^2(z)} dz = \int_{\mathbb{R}^d} \frac{e^{2\pi i \langle y - x, z \rangle}}{1 + \sum_{l=1}^{m} a_l \cdot (2\pi)^{2l} \|z\|^{2l}} dz. \tag{49}$$

D.2. Sampling Approximation

As discussed in Section 4.2, we are interested in sampling points $z \in \mathbb{R}^d$ from a finite non negative measure with density given by $w^a(z) = (1 + a \cdot (2\pi)^{2m} \|z\|^{2m})^{-1}$. With a slight overload of notation, we will also denote by w_a the scalar function $w_a : \mathbb{R} \to \mathbb{R}$,

$$w^a(r) = (1 + a \cdot (2\pi)^{2m} r^{2m})^{-1}. \tag{50}$$

First, note that $w_a(z)$ depends on z only through the norm $\|z\|$, and thus a spherically symmetric function. Therefore, with a spherical change of variables, we can rewrite the integrals w.r.t w_a^{-2} as follows: For any $f : \mathbb{R}^d \to \mathbb{C}$,

$$\int_{\mathbb{R}^d} w_a(z) f(z) dz = \int_0^\infty dr \int_{S^{d-1}} d\theta \;\; w_a(r) A_{d-1}(r) f(r\theta) \tag{51}$$

$$= A_{d-1}(1) \int_0^\infty dr \int_{S^{d-1}} d\theta \;\; \left[w_a(r) r^{d-1} \right] \cdot f(r\theta). \tag{52}$$

Here S^{d-1} the unit sphere in $\mathbb{R}^d$, θ is sampled from the uniform probability measure on the sphere, r is the radius, and

$$A_{d-1}(r) = \frac{2\pi^{d/2}}{\Gamma(d/2)} r^{d-1} \tag{53}$$

is the $d-1$ dimensional volume of the sphere or radius r in $\mathbb{R}^d$. The meaning of (52) is that to sample from w_a^{-2}, we can sample θ uniformly from the sphere (easy), and r from a density

$$\zeta(r) = w_a(r) r^{d-1} = \frac{r^{d-1}}{1 + a \cdot (2\pi)^{2m} r^{2m}} \tag{54}$$

on the real line. Note that the condition $m > d/2$ that we impose throughout is necessary. Indeed, without this condition the decay of $\zeta(r)$ would not be fast enough at infinity, and the density would not have a finite mass.

As discussed in Section 4.2, $\zeta(r)$ is a density on a real line, with a single mode and an analytic expression, which allows easy computation of the derivatives. Such distributions can be efficiently sampled using, for instance, off-the-shelf Hamiltonian Monte Carlo (HMC) samplers, (Betancourt, 2017). In our experiments we have used an even simpler scheme, by discretizing $\mathbb{R}$ into a grid of 10000 points, with limits wide enough to accommodate a wide range of parameters a.

E. A Few Basic Properties of the Kernel

Proposition 6. The kernel (11) is real valued and satisfies

$$K^a(x, y) = \int_{\mathbb{R}^d} \frac{\cos\left(2\pi \left\langle y - x, z \right\rangle\right)}{1 + a \cdot (2\pi)^{2m} \left\|z\right\|^{2m}} dz. \tag{55}$$

Proof. Write $e^{2\pi i \langle y-x, z\rangle} = cos(2\pi \langle y - x, z\rangle) + i \sin(2\pi \langle y - x, z\rangle)$ and observe that sin is odd in z, while $1 + a \cdot (2\pi)^{2m} \left\|z\right\|^{2m}$ is even. □

Proposition 7. For all $x, y \in \mathbb{R}^d$,

$$K^b(x, y) = b^{-\frac{d}{2m}} K^1(b^{-\frac{1}{2m}} x, b^{-\frac{1}{2m}} y) \tag{56}$$

Proof. Write $u = b^{\frac{1}{2m}} z$ and note that $du = (b^{\frac{1}{2m}})^d dz$. We have

$$K^b(x, y) = \int_{\mathbb{R}^d} \frac{\cos\left(2\pi \left\langle y - x, z \right\rangle\right)}{1 + b \left\|2\pi \cdot z\right\|^{2m}} dz \tag{57}$$

$$= \int_{\mathbb{R}^d} \frac{\cos\left(2\pi \left\langle b^{-\frac{1}{2m}}(y - x), u \right\rangle\right)}{1 + \left\|2\pi u\right\|^{2m}} du \cdot b^{-\frac{d}{2m}} \tag{58}$$

$$= b^{-\frac{d}{2m}} K^1(b^{-\frac{1}{2m}} x, b^{-\frac{1}{2m}} y). \tag{59}$$

□

Lemma 8. There is a function $c(m)$ of m such that for every $x \in \mathbb{R}^d$,

$$\int \left(K^a(x, y)\right)^2 dy = c(m) \cdot a^{-\frac{d}{2m}}. \tag{60}$$

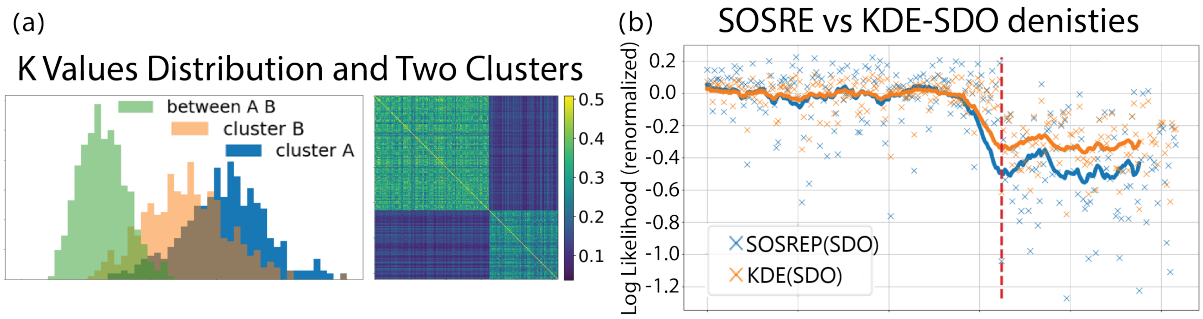

Figure 4: (a) Distribution of Kernel Values and SDO Kernel Values Inside and Between Clusters. (b) SOSREP and KDE Loglikelihoods. the x-axis represents points in the data, arranged by clusters, y-axis shows the log-likelihood.

Proof. Recall that for fixed x and a, the Fourier transform satisfies $\mathcal{F}(k_x^a)(z) = \frac{e^{-2\pi i\langle x,z\rangle}}{1+a(2\pi)^{2m}\|z\|^{2m}}$, where $k_x^a(\cdot) = K^a(x,\cdot)$ (see eq. (48)). We thus have

$$\|k_x^a\|_{L_2}^2 = \|\mathcal{F}(k_x^a)\|_{L_2}^2 \tag{61}$$

$$= \int \frac{e^{-2\pi i\langle x,z\rangle} \cdot \overline{e^{-2\pi i\langle x,z\rangle}}}{\left(1+a(2\pi)^{2m}\|z\|^{2m}\right)^2} dz \tag{62}$$

$$= \int \frac{1}{\left(1+a(2\pi)^{2m}\|z\|^{2m}\right)^2} dz \tag{63}$$

$$= a^{-\frac{d}{2m}} \int \frac{1}{\left(1+(2\pi)^{2m}\|z'\|^{2m}\right)^2} dz', \tag{64}$$

$$\tag{65}$$

where we have used the variable change $z = a^{-\frac{1}{2m}} z'$. □

F. Difference between SOSREP and KDE Models

In this Section we construct an analytic example where the SOSREP estimator may differ arbitrarily from the KDE estimator with the same kernel. Thus, the models are not equivalent, and encode different prior assumptions. Briefly, we consider a block model, with two clusters. We'll show that in this particular setting, in KDE the clusters influence each other more strongly, i.e the points in one cluster contribute to the weight of the points in other cluster, yielding more uniform models. In contrast, in SOSREP, rather surprisingly, the density does not depend on the mutual position of the clusters (in a certain sense). Note that this is not a matter of *bandwith* of the KDE, since both models use the same kernel. We believe that this property may explain the better performance of SOSREP in Anomaly Detection tasks, although further investigation would be required to verify this.

Given a set of datapoints $S = \{x_i\}$, for the purposes of this section the KDE estimator is the function

$$f_{kde}(x) = f_{kde,S}(x) = \frac{1}{|S|}\sum_i k_{x_i}(x). \tag{66}$$

Let f_{SOSREP} be the solution of (3). We will compare the ratios $f_{kde}(x_i)/f_{kde}(x_j)$ versus the corresponding quantities for SOSREP, $f^2_{SOSREP}(x_i)/f^2_{SOSREP}(x_j)$ for some pairs x_i, x_j. Note that these ratios do not depend on the normalization of f_{kde} and f^2_{SOSREP}, and can be computed from the unnormalized versions. In particular, we do not require k_{x_i} to be normalized in (66).

Consider a set S with two components, $S = S_1 \cup S_2$, with $S_1 = \{x_1, \dots, x_N\}$ and $S_2 = \{x'_1, \dots x'_M\}$ and with the following kernel values:

$$K = \begin{cases} k(x_i, x_i) = k(x'_j, x'_j) = 1 & \text{for all } i \le N, j \le M \\ k(x_i, x_j) = \gamma^2 & \text{for } i \neq j \\ k(x'_i, x'_j) = \gamma'^2 & \text{for } i \neq j \\ k(x_i, x'_j) = \beta\gamma\gamma' & \text{for all } i, j \end{cases} \tag{67}$$

This configuration of points is a block model with two components, or two clusters. The correlations between elements in the first cluster are γ^2, and are γ'^2 in the second cluster. Inter-cluster correlations are $\beta\gamma\gamma'$. We assume that $\gamma, \gamma, \beta \in [0, 1]$ and w.l.o.g take $\gamma > \gamma'$. While this is an idealized scenario to allow analytic computations, settings closely approximating the configuration (67) often appear in real data. See Section F.1 for an illustration. In particular, Figure 4 show a two cluster configuration in that data, and the distribution of $k(x, x')$ values.

The KDE estimator for K is simply

$$f_{kde}(x_t) = \frac{1}{N+M}\left[1 + (N-1)\gamma^2 + M\beta\gamma\gamma'\right] \approx \frac{N}{N+M}\gamma^2 + \frac{M}{N+M}\beta\gamma\gamma', \tag{68}$$

for $x_t \in S_1$, where the second, approximate equality, holds for large M, N. To simplify the presentation, we shall use this approximation. However, all computations and conclusions also hold with the precise equality. For $x'_t \in S_2$ we similarly have $f_{kde}(x'_t) \approx \frac{N}{N+M}\beta\gamma\gamma' + \frac{M}{N+M}\gamma'^2$, and when $M = N$, the density ratio is

$$\frac{f_{kde}(x_t)}{f_{kde}(x'_t)} = \frac{\gamma^2 + \beta\gamma\gamma'}{\gamma'^2 + \beta\gamma\gamma'}. \tag{69}$$

The derivation of the SOSREP estimator is considerably more involved. Here we sketch the argument, while full details are given in Supplementary Material Section G. First, recall from the previous section that the natural gradient in the α coordinates is given by $2\left(\beta - N^{-1}(K\beta)^{-1}\right)$. Since the optimizer of (3) must satisfy $\nabla_f L = 0$, we are looking for $\beta \in \mathbb{R}^{N+M}$ such that $\beta = (K\beta)^{-1}$ (the term N^{-1} can be accounted for by renormalization). Due to the symmetry of K and since the minimizer is unique, we may take $\beta = (a, \dots, a, b, \dots, b)$, where a is in first N coordinates and b is in the next M. Then $\beta = (K\beta)^{-1}$ is equivalent to a, b solving the following system:

$$\begin{cases} a &= a^{-1}\left[1 + (N-1)\gamma^2\right] + b^{-1}M\beta\gamma\gamma' \\ b &= a^{-1}N\beta\gamma\gamma' + b^{-1}\left[1 + (M-1)\gamma'^2\right] \end{cases} \tag{70}$$

This is a non linear system in a, b. However, it turns out that it may be explicitly solved, up to a knowledge of a certain sign variable (see Proposition 10). Moreover, for $M = N$, the dependence on that sign variable vanishes, and we obtain

Proposition 9. Consider the kernel and point configuration described by (67), with $M = N$. Then for every $x_t \in S_1, x'_s \in S_2$,

$$\frac{f_{SOSREP}(x_t)}{f_{SOSREP}(x'_s)} = \frac{\gamma^2}{\gamma'^2}. \tag{71}$$

In particular, the ratio does not depend on β.

It remains to compare the ratio (71) to KDE's ratio (69). If $\beta = 0$, when the clusters are maximally separated, the ratios coincide. However, let us consider the case, say, $\beta = \frac{1}{2}$, and assume that $\gamma' \ll \gamma$. Then in the denominator of (69) the larger term is $\beta\gamma\gamma'$, which comes from the influence of the first cluster on the second. This makes the whole ratio to be of the order of a constant. On the other hand, in SOSREP there is no such influence, and the ratio (71) may be arbitrarily large. We thus expect the gap between the cluster densities to be larger for SOSREP, which is indeed the case empirically. One occurence of this on real data is illustrated in Figure 4.

F.1. Evaluation of the Difference Between SOSREP and KDE for Real Data

We have performed spectral clustering of the "letter" dataset from ADBech ((Han et al., 2022)), using the empirical SDO kernel as affinity matrix for both SOSREP and KDE. We then have chosen two clusters that most resemble the two block

model (67) in Section F. The kernel values inside and between the clusters are shown in Figure 4a. Next, we train the SOSREP and KDE models for just these two clusters (to be compatible with the setting of Section F. The results are similar for densities trained on full data). The log of these SOSREP and KDE densities in shown in Figure 4b (smoothed by running average). By adding an appropriate constant, we have arranged that the mean of both log densities is 0 on the first cluster. Then one can clearly see that the gap between the values on the first and second cluster is larger for the SOSREP model, yielding a less uniform model, as expected from the theory.

G. KDE vs SOSREP Comparison Proofs

In this section we develop the ingredients required to prove Proposition 9. In section G.1 we reduce the solution of the SOSREP problem for the two block model to a solution of a non-linear system in two variables, and derive the solution of this system. In section G.2 we use these results to prove Proposition 9.

G.1. Solution Of SOSREP for a 2-Block Model

As discussed in section F, any SOSREP solution f must be a zero point of the natural gradient, $\nabla_f L = 0$. Using the expressions given following Lemma 1, this implies $\beta = \frac{1}{N}(K\beta)^{-1}$. Since we are only interested in f up to a scalar normalization, we can equivalently assume simply $\beta = (K\beta)^{-1}$. Further, by symmetry consideration we may take $\beta = (a, \dots, a, b, \dots, b)$, where a is in first N coordinates and b is in the next M. Then, as mentioned in section F, $\beta = (K\beta)^{-1}$ is equivalent to a, b solving the following system:

$$\begin{cases} a &= a^{-1}\left[1 + (N-1)\gamma^2\right] + b^{-1}M\beta\gamma\gamma' \\ b &= a^{-1}N\beta\gamma\gamma' + b^{-1}\left[1 + (M-1)\gamma'^2\right] \end{cases} \tag{72}$$

It turns out that it is possible to derive an expression for the ratio of the squares of the solutions to this system in the general case.

Proposition 10 (Two Variables SOSREP System). Let a, b be solutions of

$$\begin{cases} a &= H_{11}a^{-1} + H_{12}b^{-1} \\ b &= H_{21}a^{-1} + H_{22}b^{-1} \end{cases} \tag{73}$$

Then

$$a^2/b^2 = H_{11}^2 \left(\frac{-(H_{21} + H_{12}) - \rho\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}{2H_{11}H_{22} + H_{12}\left[-(H_{21} - H_{12}) + \rho\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}\right]} \right)^2 \tag{74}$$

for a ρ satisfying $\rho \in \{+1, -1\}$.

Proof. Write $u = a^{-1}$, $v = b^{-1}$, and multiply the first and second equations by u and v respectively. Then we have

$$\begin{cases} 1 &= H_{11}u^2 + H_{12}uv \\ 1 &= H_{21}uv + H_{22}v^2. \end{cases} \tag{75}$$

We write

$$v = \left(1 - H_{11}u^2\right)/H_{12}u. \tag{76}$$

Also, from the first equation,

$$H_{12}uv = 1 - H_{11}u^2. \tag{77}$$

Substituting into the second equation,

$$1 = \frac{H_{21}}{H_{12}}\left(1 - H_{11}u^2\right) + H_{22}\frac{\left(1 - H_{11}u^2\right)^2}{\left(H_{12}u\right)^2}. \tag{78}$$

Finally setting $s = u^2$ and multiplying by $H_{12}^2 s$,

$$H_{12}^2 s = H_{21} H_{12} s \left(1 - H_{11} s\right) + H_{22} \left(1 - H_{11} s\right)^2 . \tag{79}$$

Collecting terms, we have

$$s^2 (H_{11}^2 H_{22} - H_{11} H_{12} H_{21}) + s(H_{12} H_{21} - H_{12}^2 - 2H_{11} H_{22}) + H_{22} = 0. \tag{80}$$

Solving this, we get

$$s = \frac{-(H_{12} H_{21} - H_{12}^2 - 2H_{11} H_{22}) \pm \sqrt{(H_{12} H_{21} - H_{12}^2 - 2H_{11} H_{22})^2 - 4(H_{11}^2 H_{22} - H_{11} H_{12} H_{21}) H_{22}}}{2(H_{11}^2 H_{22} - H_{11} H_{12} H_{21})}. \tag{81}$$

The expression inside the square root satisfies

$$\begin{aligned}
&(H_{12} H_{21} - H_{12}^2 - 2H_{11} H_{22})^2 - 4(H_{11}^2 H_{22} - H_{11} H_{12} H_{21}) H_{22} && (82)\\
&= (H_{12}(H_{21} - H_{12}) - 2H_{11} H_{22})^2 - 4(H_{11}^2 H_{22} - H_{11} H_{12} H_{21}) H_{22} && (83)\\
&= H_{12}^2 (H_{21} - H_{12})^2 - 4H_{11} H_{22} H_{12} (H_{21} - H_{12}) + 4H_{11} H_{12} H_{21} H_{22} && (84)\\
&= H_{12}^2 (H_{21} - H_{12})^2 + 4H_{11} H_{22} H_{12}^2 && (85)\\
&= H_{12}^2 \left[(H_{21} - H_{12})^2 + 4H_{11} H_{22}\right] && (86)
\end{aligned}$$

Thus, simplifying, we have

$$u^2 = s = \frac{-(H_{12}(H_{21} - H_{12}) - 2H_{11} H_{22}) + \rho H_{12} \sqrt{(H_{21} - H_{12})^2 + 4H_{11} H_{22}}}{2H_{11}(H_{11} H_{22} - H_{12} H_{21})}, \tag{87}$$

where $\rho \in \{+1, -1\}$.

Rewriting (76) again, we have

$$v^2 = \frac{\left(1 - H_{11} u^2\right)^2}{H_{12}^2 u^2}. \tag{88}$$

Further,

$$a^2/b^2 = v^2/u^2 = \frac{\left(1 - H_{11}u^2\right)^2}{H_{12}^2 u^4} \tag{89}$$

$$= \left(\frac{1 - H_{11}u^2}{H_{12}u^2}\right)^2 \tag{90}$$

$$= \left(\frac{1}{H_{12}u^2} - \frac{H_{11}}{H_{12}}\right)^2 \tag{91}$$

$$= \left(\frac{H_{11}}{H_{12}}\right)^2 \left(\frac{2(H_{11}H_{22} - H_{12}H_{21})}{-(H_{12}(H_{21} - H_{12}) - 2H_{11}H_{22}) + \rho H_{12}\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}} - 1\right)^2 \tag{92}$$

$$= \left(\frac{H_{11}}{H_{12}}\right)^2 \left(\frac{2(H_{11}H_{22} - H_{12}H_{21}) + (H_{12}(H_{21} - H_{12}) - 2H_{11}H_{22}) - \rho H_{12}\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}{-(H_{12}(H_{21} - H_{12}) - 2H_{11}H_{22}) + \rho H_{12}\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}\right)^2 \tag{93}$$

$$= \left(\frac{H_{11}}{H_{12}}\right)^2 \left(\frac{-2H_{12}H_{21} + H_{12}(H_{21} - H_{12}) - \rho H_{12}\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}{-(H_{12}(H_{21} - H_{12}) - 2H_{11}H_{22}) + \rho H_{12}\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}\right)^2 \tag{94}$$

$$= H_{11}^2 \left(\frac{-2H_{21} + H_{21} - H_{12} - \rho\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}{-(H_{12}(H_{21} - H_{12}) - 2H_{11}H_{22}) + \rho H_{12}\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}\right)^2 \tag{95}$$

$$= H_{11}^2 \left(\frac{-(H_{21} + H_{12}) - \rho\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}{-(H_{12}(H_{21} - H_{12}) - 2H_{11}H_{22}) + \rho H_{12}\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}\right)^2 \tag{96}$$

$$= H_{11}^2 \left(\frac{-(H_{21} + H_{12}) - \rho\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}{2H_{11}H_{22} + H_{12}\left[-(H_{21} - H_{12}) + \rho\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}\right]}\right)^2 \tag{97}$$

□

G.2. Proof Of Proposition 9

Similarly to the case with KDE, we will use the following approximation of the system (72)

$$\begin{cases} a &= a^{-1}N\gamma^2 + b^{-1}M\beta\gamma\gamma' \\ b &= a^{-1}N\beta\gamma\gamma' + b^{-1}M\gamma'^2 \end{cases} \tag{98}$$

Proof. Let f be the SOSREP solution. By definition, the ratio $\frac{f(x_t)}{f(x_s')}$ is given by a^2/b^2 where a, b are the solutions to (98). That is, we take $H_{12} = H_{21} = \beta\gamma\gamma'$, $H_{11} = \gamma^2$, and $H_{22} = \gamma'^2$ in Proposition 10. Note that we have removed the dependence on N, since it does not affect the ratio. By Proposition 10, substituting into (74),

$$H_{11}^2 \left(\frac{-(H_{21} + H_{12}) - \rho\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}}{2H_{11}H_{22} + H_{12}\left[-(H_{21} - H_{12}) + \rho\sqrt{(H_{21} - H_{12})^2 + 4H_{11}H_{22}}\right]}\right)^2 \tag{99}$$

$$= H_{11}^2 \left(\frac{-H_{12} - \rho\sqrt{H_{11}H_{22}}}{H_{11}H_{22} + H_{12}\rho\sqrt{H_{11}H_{22}}}\right)^2 \tag{100}$$

$$= \gamma^4 \left(\frac{-2\beta\gamma\gamma' - 2\rho\gamma\gamma'}{2\gamma^2\gamma'^2 + 2\beta\gamma\gamma'\rho\gamma\gamma'}\right)^2 \tag{101}$$

$$= \left(\frac{\gamma^2\gamma\gamma'}{\gamma^2\gamma'^2}\right)^2 \frac{(\beta + \rho)^2}{(1 + \beta\rho)^2} \tag{102}$$

It remains to note that $\frac{(\beta+\rho)^2}{(1+\beta\rho)^2} = 1$ for any β and $\rho \in \{+1, -1\}$. □

H. Invariance of $\mathcal{C}$ under Natural Gradient

Define the non-negative cone of functions $\mathcal{C} \subset \mathcal{H}$ by

$$\mathcal{C} = \{f \in \mathcal{H} \mid f(x) \geq 0 \ \ \forall x \in \mathcal{X}\}. \tag{103}$$

As discussed in section 3.1, the functional $L(f)$ is convex on $\mathcal{C}$.

We now show that if the kernel k is non-negative, then the cone $\mathcal{C}$ is invariant under the natural gradient steps. In particular, this means that if one starts with initialization in $\mathcal{C}$ (easy to achieve), then the optimization trajectory stays in $\mathcal{C}$, without a need for computationally heavy projection methods. Note that this is unlikely to be true for the standard gradient. Recall that the expression (6) for the natural gradient is given in Lemma 1.

Proposition 11. Assume that $k(x, x') \geq 0$ for all $x, x' \in \mathcal{X}$ and that $\lambda < 0.5$. If $f \in \mathcal{C}$, then also $f' := f - 2\lambda \left[f - \frac{1}{N} \sum_{i=1}^{N} f^{-1}(x_i) k_{x_i} \right] \in C$.

Proof. Indeed, by opening the brackets,

$$f' = (1 - 2\lambda) f + 2\lambda \left[\frac{1}{N} \sum_{i=1}^{N} f^{-1}(x_i) k_{x_i} \right],$$

which is a non-negative combination of functions in $\mathcal{C}$, thus yielding the result. □

I. Fisher Divergence for Hyper-Parameters Selection

The handling of unnormalized models introduces a particular nuance in the context of hyperparameter tuning, as it prevents the use of the maximum likelihood of the data in order to establish the optimal parameter. When confronted with difficulties associated with normalization, it is common to resort to score-based methods. The score function is defined as

$$s(x; a) = \nabla_x \log p_m(x; a), \tag{104}$$

where $p_m(x; a)$ is a possibly unnormalized probability density on $\mathbb{R}^d$, evaluated at $x \in \mathbb{R}^d$, and dependent on the hyperparameter a. Since the normalization constant is independent of x, and s is defined via the gradient in x, s is independent of the normalization. As a result, distance metrics between distributions that are based on the score function, such as the Fisher Divergence, can be evaluated using non-normalized distributions.

The Fisher Divergence (FD) is a similarity measure between distributions, which is based on the score function – the gradient of the log likelihood. In particular, it does not require the normalization of the density. While the link between FD and maximum likelihood estimation has been previously studied in the context of score matching (Lyu, 2012), to the best of our knowledge, this represents the first application of FD for hyperparameter tuning.

The divergence between data and a model can be approximated via the methods of (Hyvärinen, 2005), which have been recently computationally improved in (Song et al., 2020) in the context of score-based generative models, (Song and Ermon, 2019). In particular, we adapt the Hutchinson trace representation-based methods used in (Song et al., 2020) and (Grathwohl et al., 2018) to the case of models of the form (3).

In this work, we employ this concept, leveraging it to identify the choice of parameters (in our case, a, the smoothness parameter) that minimize the FD between the density learned on the training set and the density inferred on the test set. Specifically, we apply *score-matching* ((Hyvärinen, 2005)), a particular approach to measuring the Fisher divergence between a dataset sampled from an unknown distribution and a proposed distribution model.

I.1. Score Matching and Fisher Divergence

Given independent and identically distributed samples $x_1, \ldots, x_N \in \mathbb{R}^D$ from a distribution $p_d(x)$ and an un-normalized density learned, $\tilde{p_m}(x; a)$ (where a is a parameter). Score matching sets out to reduce the Fisher divergence between p_d and $\tilde{p_m}(\cdot; a)$, formally expressed as

$$L(a) = \frac{1}{2} \cdot E_{p_d}[\|s_m(x; a) - s_d(x)\|^2]$$

As detailed in (Hyvärinen, 2005), the technique of integration by parts can derive an expression that does not depend on the unknown latent score function s_d:

$$L(a; x_1, \ldots, x_n) = \frac{1}{N} \sum_{i=1}^{N} \left[tr(\nabla_x s_m(x_i; a)) + \frac{1}{2} \cdot \|s_m(x_i; a)\|^2 \right] + C$$

In this context, C is a constant independent of a, $tr(\cdot)$ denotes the trace of a matrix, and $\nabla_x s_m(x_i; a) = \nabla_x^2 log(\tilde{p_m}(x_i; a))$ is the Hessian of the learned log-density function evaluated at x_i.

I.2. The Hessian Estimation for Small a's

Deriving the Hessian for small a values proves to be challenging. Note that small a values signify overfitting to the training data, consequently, this leads to a density that is mainly close to zero between samples, thereby making the process highly susceptible to significant errors in numerically calculating the derivatives. This situation results in a Hessian that is fraught with noise. Hence, our strategy focuses on locating a stable local minimum with the highest possible a. In this context, we define a stable local minimum as a point preceded and succeeded by three points, each greater than the focal point.

I.3. Approximating the Hessian Trace

Although this method holds promise, it's worth noting the computational burden tied to the calculation of the Hessian trace. To mitigate this, we rely on two techniques. First, we utilize Hutchinson's trace estimator (Hutchinson, 1989), a robust estimator that facilitates the estimation of any matrix's trace through a double product with a random vector ϵ:

$$Tr(H) = E_\epsilon \left[\epsilon^T H \epsilon \right].$$

Here ϵ is any random vector on $\mathbb{R}^d$ with mean zero and covariance I. This expression allows to reduce amount of computation of $Tr(H)$, by computing the products $H\epsilon$ directly, for a few samples of ϵ, without the need to compute the full H itself. A similar strategy has been recently employed in (Grathwohl et al., 2018) in a different context, for a trace computation of a Jacobian of a density transformation, instead of the score itself.

In more detail, score computations can be performed efficiently and in a 'lazy' manner using automatic differentiation, offered in frameworks such as PyTorch. This allows us to compute a vector-Hessian product $H\epsilon$ per sample without having to calculate the entire Hessian for all samples, a tensor of dimensions $N \times (d \times d)$, in advance. More specifically, we utilize PyTorch's automatic differentiation for computing the score function, which is a matrix of $N \times d$. Subsequently, this is multiplied by ϵ. We then proceed with a straightforward differentiation $\nabla_x s(x_i) = \frac{1}{h} \cdot (s(x_i + h \cdot \epsilon) - s(x_i))$ for small step h, followed by a summation which is lazily calculated through PyTorch (see Algorithm 1).

Algorithm 1 Calculating Hutchinson's Trace Estimator

Require: Score function s, small constant h, sample x, # of random vectors n
1: Initialize $traceEstimator$ to 0
2: **for** $i = 1$ to n **do**
3: Sample random vector ϵ from normal distribution
4: Calculate $s(a; x + h * \epsilon)$
5: Calculate $(s(a; x + h * \epsilon) - s(a; x))$
6: Compute $(1/h) \cdot (s(a; x + h * \epsilon) - s(a; x)) \cdot \epsilon$
7: Add result to $traceEstimator$
8: **end for**
9: Return $\frac{traceEstimator}{n}$

J. Experiments

J.1. Evaluation on 2D synthetic dataset with different kernels

This section presents the application of SOSREP using different kernels on a 2D toy dataset, following the methodologies outlined in (Song and Ermon, 2019). Through these comparisons, we aim to enhance our understanding of what makes SOSREP particularly effective, especially with the SDO kernel.

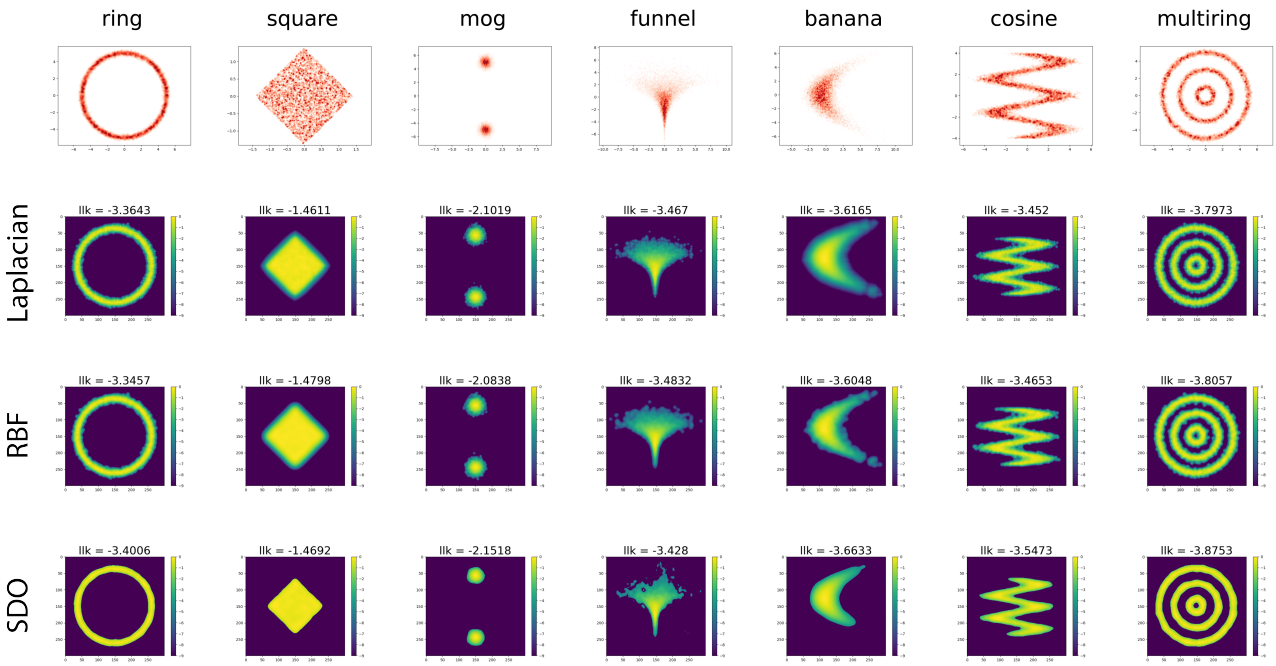

Figure 5: Log densities learned by SOSREP with the Laplacian, Gaussian, and SDO kernels. Columns are different synthetic datasets. For visualization, We subtracted the maximum from each log density and clipped the minimum value at -9, following (Song and Ermon, 2019). Above each panel are shown the average log-likelihoods on a test set.

In Figure 5, we evaluate the SOSREP estimator using SDO, Gaussian, and Laplacian kernels on a number of 2D datasets. These datasets were also used in (Song and Ermon, 2019) (Figure 2) to evaluate various state-of-the-art density estimation methods. The top row in Figure 5 displays the ground truth data, with columns representing different datasets and rows corresponding to the various kernels. The color intensity indicates the values of the log-likelihood. Additionally, the average log-likelihood (llk) was computed on a test set, details of which are provided below.

We adhered closely to the experimental protocol from (Song and Ermon, 2019), using their code to generate data. Exceptions were the Mixture of Gaussians (MOG) and Square datasets, for which we developed our own code, as original code was not available. All datasets were generated with $N = 10000$ samples. Like the method studied in (Song and Ermon, 2019), SOSREP is not normalized. For log-likelihood computation (llk), the normalization constant was estimated using an importance sampling procedure detailed in (Song and Ermon, 2019), Section 3.2, which we replicated for our experiments in Figure 2. In 2D, such approximations are reasonably effective. For clarity in visualization, the maximum log-likelihood was subtracted (setting the maximum llk to 0) and clipped at -9. This adjustment was made solely for visualization purposes and not for the llk computation. The bandwidth parameter a for SOSREP was selected based on the best LLK from a verification set using a grid search over a grid of 10 values.

As shown in Figure 2 and in comparison to Figure 2 of (Song and Ermon, 2019), SOSREP is one of the few methods that consistently produces clean fits across all datasets. The only other method achieving similar results is "DKEF-G-15" and, to a lesser extent, "DKEF-G".

It is noteworthy that almost all methods in Figure 2 of (Song and Ermon, 2019) assign a relatively high likelihood to the center of the multi-ring dataset, whereas SOSREP shows a distinct drop in likelihood at this point. Moreover, SOSREP accurately captures the sharp edges in the square and cosine datasets.

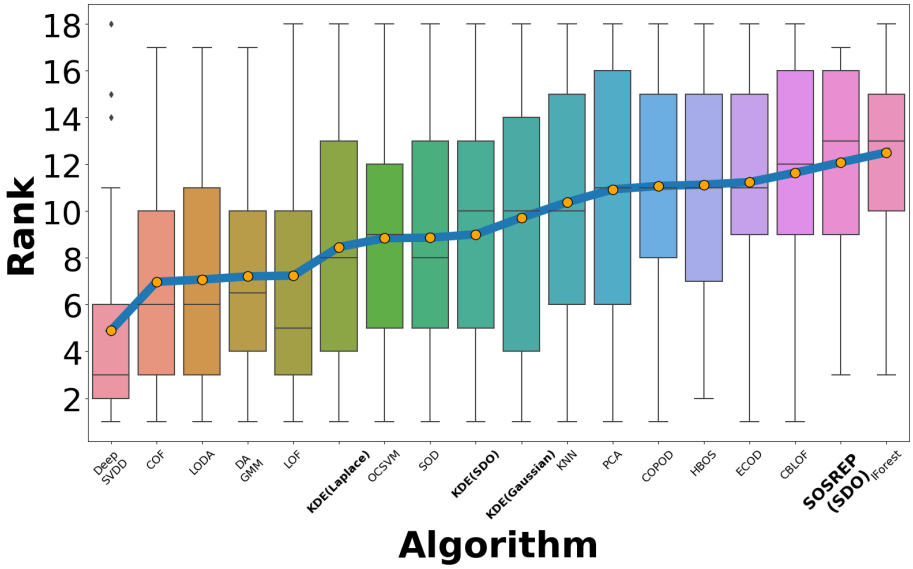

Figure 6: Anomaly Detection Results on ADBench. Relative Ranking Per Dataset, Higher is Better. SOSREP is Second Best Among 18 Algorithms

J.2. AUC-ROC Performance Analysis

In addition to AUC-ROC, we focus on a ranking system where raw AUC-ROC scores from each dataset are converted into rankings from 1 to 18, with 18 indicating the best performance. This approach reduces bias from averaging AUC-ROC scores across datasets, particularly as scores tend to be higher on easier datasets. This is crucial as no single AD method consistently excels across all scenarios, a point elaborated in (Han et al., 2022).

In Figure 6, each algorithm is represented by a box plot. The average AUC-ROC ranking across all datasets is marked by an orange dot. The 25th and 75th percentiles define the box boundaries, while the median is indicated by the internal line. Algorithms are ordered based on the average AUC-ROC ranking.

Figure 7 presents a heatmap of the AUC-ROC values. In this visualization, circle sizes correlate with the AUC values and the color gradient reflects deviations from SOSREP's performance, providing a graphical representation of how AUC-ROC levels compare with those of SOSREP.

J.3. Duplicate Anomalies.

Duplicate anomalies are often encountered in various applications due to factors such as recording errors (Kwon et al., 2018), a circumstance termed as "anomaly masking" (Campos et al., 2016; Guha et al., 2016), posing significant hurdles for diverse AD algorithms. The significance of this factor is underscored in ADbench (Han et al., 2022), where duplicate anomalies are regarded as the most difficult setup for anomaly detection, thereby attracting considerable attention. To replicate this scenario, ADbench duplicates anomalies up to six times within training and test sets, subsequently examining the ensuing shifts in AD algorithms' performance across the 47 datasets discussed in our work.

As shown in ADBench, unsupervised methods are considerably susceptible to repetitive anomalies. In particular, as shown in Fig. 7a in the main text there, performance degradation is directly proportional to the increase in anomaly duplication. With anomalies duplicated six times, unsupervised methods record a median AUC-ROC reduction of -16.43%, where in SOSREP the drop is less then 2%. This universal performance deterioration can be attributed to two factors: 1) The inherent assumption made in these methods that anomalies are a minority in the dataset, a presumption crucial for detection. The violation of this belief due to the escalation in duplicated anomalies triggers the noticed performance downturn. 2) Methods based on nearest neighbours assume that anomalies significantly deviate from the norm (known as "Point anomalies"). However, anomaly duplication mitigates this deviation, rendering the anomaly less distinguishable. Notice that while the first factor is less of a problem for a density based AD algorithm (any time the anomalies are still not the major part of the data), the second factor could harmful to DE based AD algorithms as well. The evidence of SOSREP possessing the highest median and its robustness to duplicate anomalies, along with a probable resistance to a high number of anomalies, not only

Figure 7: Heatmap of AUC-ROC values. Circles size represents absolute value, and the color the shift from SOSREP.

emphasizes its superiority as a DE method but also underscores its potential to serve as a state-of-the-art AD method.

J.4. Comparison of SOSREP for Different Kernels

In this section, we evaluate the SOSREP density estimator with various kernels on the ADBench task. Specifically, we consider the SDO kernel given by (11), the L_2 Laplacian kernel, and the Gaussain kernel. The Laplacian (aka Exponential) kernel is given by

$$k(x, y) = (1/\sigma^d) \cdot e^{-\|x-y\|/\sigma} \tag{105}$$

for $\sigma > 0$, where $\|x - y\|$ is the Euclidean norm on $\mathbb{R}^d$. The Gaussian kernel is given by

$$k(x, y) = (1/\sigma^d) \cdot e^{-\|x-y\|^2/(2\sigma^2)}. \tag{106}$$

Generally, in this task, we observe a similar behavior across all kernels, although the Gaussian kernel underperforms on a few datasets.

Figure 17 presents the performance of each kernel on the ADBench benchmark (see section 6.1). Specifically, we plot the test set AUC values on the Y-axis against the different datasets on the X-axis. Note that due to convexity, the results are nearly identical for different initializations. Figures 16 and 15 display the ADbench results in a manner consistent with Section 6.1. It is noteworthy that the Laplacian kernel also achieves impressive results. Furthermore, when considering ranks and thus mitigating the impact of extreme AUC values, the Gaussian kernel demonstrates significantly improved performance.

J.5. Per-dataset comparison of SOSREP for different kernels

Figure 12 presents the AUC-ROC results for SOSREP with the SDO kernel, compared to other algorithms in ADBench. The datasets are displayed on the x-axis. For each dataset, the figure shows a box plot of the quantity

$$\text{(AUC-ROC SOSREP)} - \text{(AUC-ROC other algorithm)} \tag{107}$$

where "other algorithm" varies over 17 other benchmark algorithms from ADBench. Positive values indicate an advantage in performance for SOSREP. The datasets are ordered by the AUC-ROC of SOSREP.

Figures 13 and 14 present the same plots for the SOSREP with the Gaussian and Laplacian kernels, respectively.

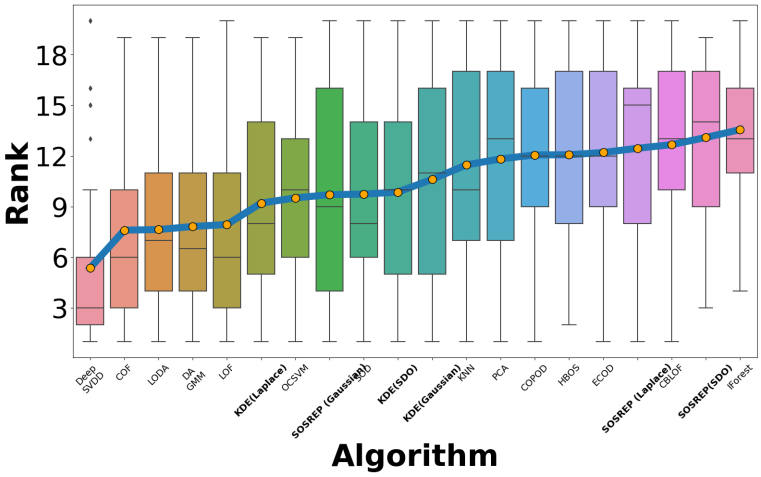

Figure 8: Anomaly Detection Results on ADBench. Mean Relative Ranking Per Dataset, Higher is Better.

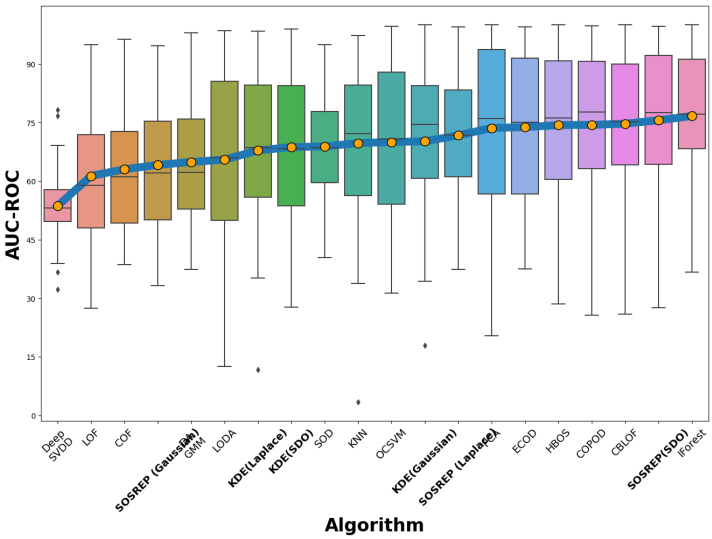

Figure 9: Anomaly Detection Results on ADBench. Mean AUC Per Dataset, Higher is Better.

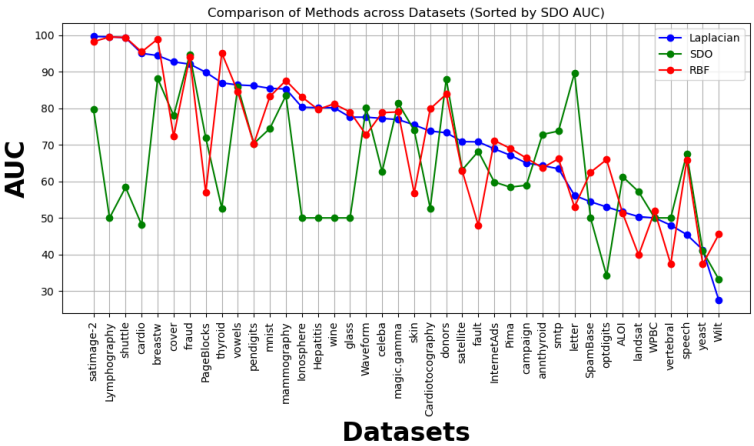

Figure 10: Comparing SOSREP results on ADBench for different kernels.

Figure 11: Evaluating SOSREP with various kernels on the ADbench benchmark.

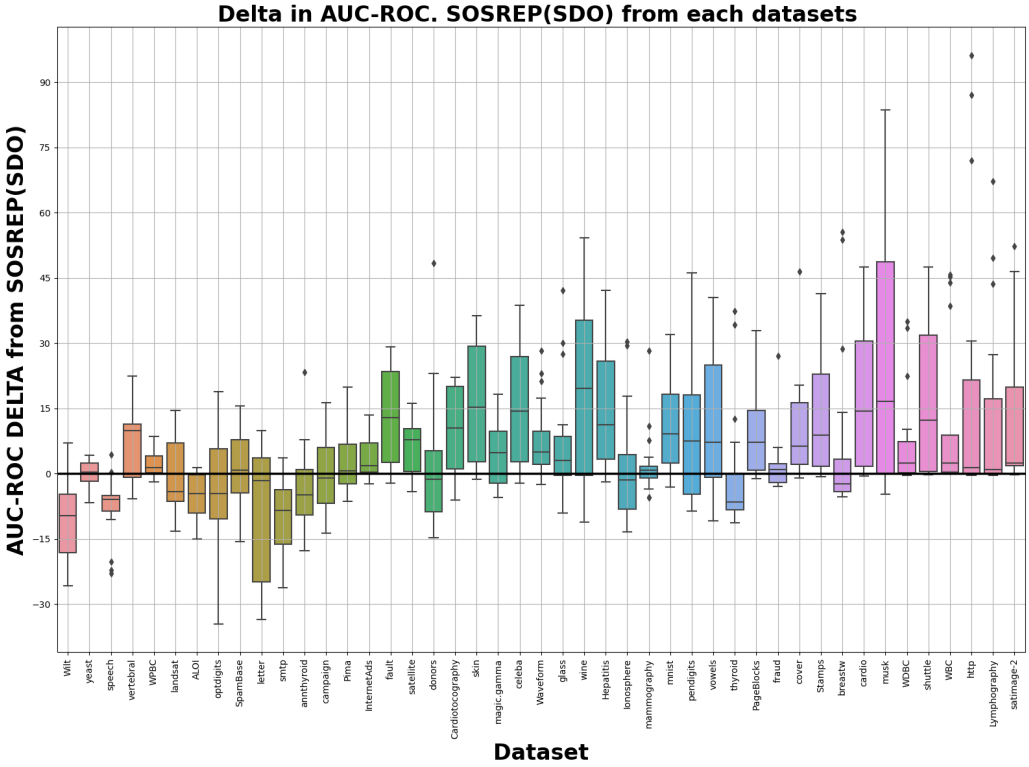

Figure 12: AUC-ROC results for SOSREP with the SDO kernel compared to the other algorithms in ADBench.

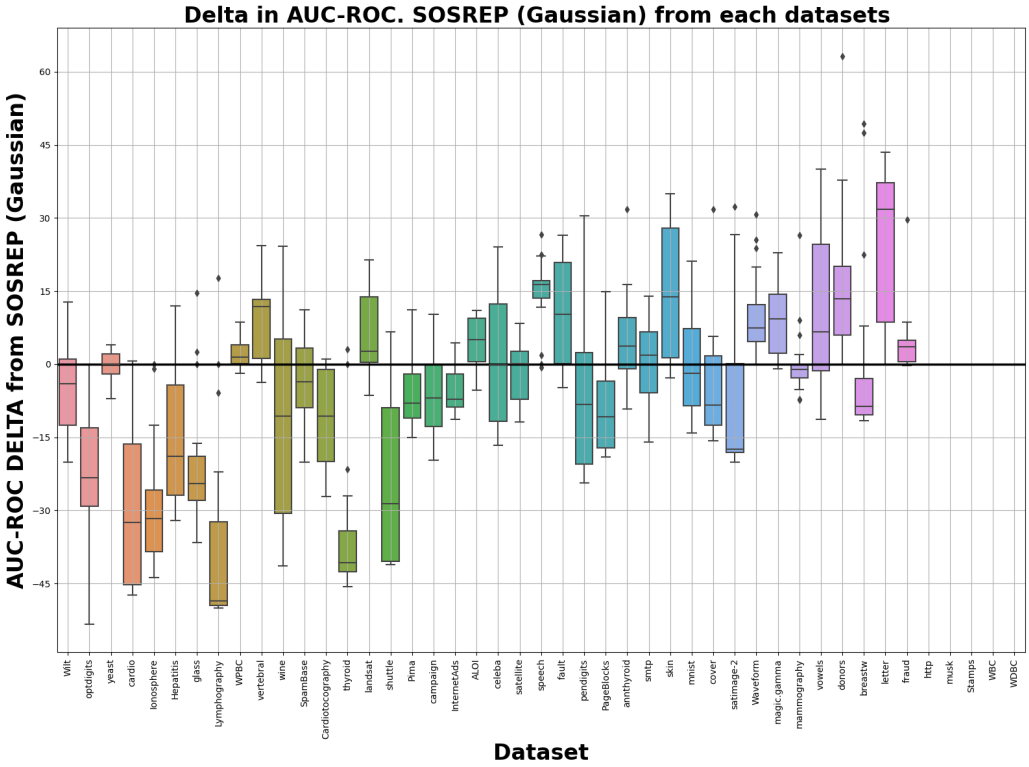

Figure 13: AUC-ROC results for SOSREP with the Gaussian kernel compared to the other algorithms in ADBench.

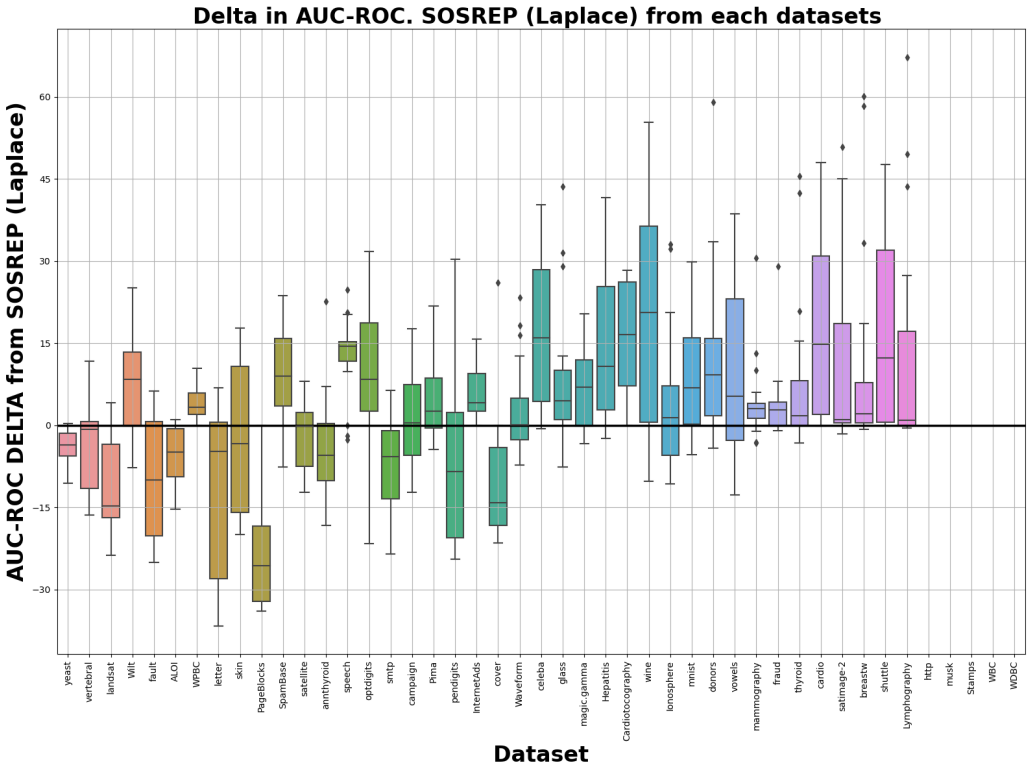

Figure 14: AUC-ROC results for SOSREP with the Laplace kernel compared to the other algorithms in ADBench.

J.6. Hyperparameter Tuning with On-the-fly Fisher-Divergence Minimization

A primary task at hand involves hyperparameter tuning to select the optimal a according to the Fisher-Divergence (FD) procedure, as detailed in section I and Section 6.1. For efficient computation, we employ an on-the-fly calculation approach. The process initiates with a a value that corresponds to the 'order'-th largest tested point. Subsequently, we explore one larger and one smaller a value. If both these points exceed the currently tested point, we continue sampling. Otherwise, we shift the tested point to a smaller value. To avoid redundant calculations, the results are continually stored, ensuring each result is computed only once. This methodology provides a balance between computational efficiency and thorough exploration of the hyperparameter space.

K. Algorithm Overview

In this section, we outline the procedure for our density estimation algorithm. Let $X \in \mathbb{R}^{N\times d}$ represent the training set and $Y \in \mathbb{R}^{N\times d}$ denote the test set for which we aim to compute the density. Initially, we establish the following hyper-parameters: 1.N_z, the number of samples Z taken for the kernel estimation. (Notice N_z is T from (12)) 2.n_{iters}, the number of iterations used in the gradient decent approximating the optimization for $\boldsymbol{\alpha}$. 3.lr, the learning rate applied in the gradient decent $\boldsymbol{\alpha}$ optimization process. 4.n_{fd_iters}, the number of iterations for the Hessian trace approximation. 5.h, the step size employed for the Hessian trace. Then, we follow Algorithm 2.

L. Figure 3 - List of Datasets

Those are the datasets for which we compared the fractions of negative values for $alpha$ optimization vs natural gradients presented in Figure 3, ordered according to the X-axis in the figure :

1. Mnist
2. Shuttle

Algorithm 2 Density Estimation Procedure with Hypeparameter Tuning

repeat
  Estimate the SDO kernel matrix via sampling as described in Section 3 and detailed in Algorithm 3.
  Determine the optimal $\boldsymbol{\alpha}$ as described in Section 3 and detailed in Algorithm 4. $optimal_\alpha$ forms $f := f^* \triangleq f_{optimal_\alpha}$ and the density estimator is $F^2 \triangleq (f)^2$.
  Compute $F^2(Y)$ the density over Y as described in Section 3 and detailed in Algorithm 5.
  Assess the Fisher-divergence as described in Supplementary Material Section I and detailed in both Algorithm 6 and Algorithm 1.
until For each smoothness parameter a.
The density estimator F^2 corresponds to the a value that yields the minimal FD.

Algorithm 3 $K(\cdot) \to \mathbb{R}^{2\times N}$: Sampling the Multidimensional SDO Kernel

Require: X, Y, a
$\boldsymbol{\theta} \leftarrow \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2}$; $\boldsymbol{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
$\boldsymbol{r} \sim \frac{r^{d-1}}{1+a(2\pi r)^{2m}}$ using grid search.
$Z \leftarrow \boldsymbol{r} \cdot \boldsymbol{\theta}^T$
$\boldsymbol{b} \sim U[0, 2\pi]$
samples $\leftarrow \frac{1}{N_z} \cdot \cos(X \cdot Z + b) \cdot (\cos(Y \cdot Z + b))^T$

Algorithm 4 $Optimal_alpha(\cdot) \to \mathbb{R}^N$: Calculating the Optimal Alphas

Require: $X, \text{lr}, n_{\text{iters}}$
$K \leftarrow K(X, X)$
$\alpha \leftarrow [|\alpha_0|, \ldots, |\alpha_N|] : \alpha_i \sim \mathcal{N}(0, 1)$
for $i = 1$ in n_{iters} **do**
  $\alpha \leftarrow \alpha - 2 \cdot lr \cdot ((K \cdot \alpha) - (K \cdot (1./(K \cdot \alpha)))/N_{\text{data}})$
end for

Algorithm 5 $F^2(\cdot) \to \mathbb{R}^{N_Y}$: Density over coordinates Y given observations X.

Require: X, Y
$\alpha \leftarrow Optimal_alpha(X, \ldots)$
$e \leftarrow K(X, Y)$
$density \leftarrow (e \cdot \alpha)^2$

Algorithm 6 $FD(\cdot) \to \mathbb{R}$: Fisher Divergence with Hessian Trace Approximation.

Require: X(Train set), Y(Test set), n_{fd_iters}, h
$scores \leftarrow \nabla \sum \log(F^2(X, Y))$
traces_sum $\leftarrow \mathbf{0}$
for $i = 1$ in n_{iters} **do**
  $\epsilon \sim \mathcal{U}(\{-1, 0, 1\}^{\text{size}(X)})$
  shifted_scores $\leftarrow \nabla \sum \log(F^2(X, Y + h \cdot \epsilon))$
  traces_sum $\leftarrow$ traces_sum $+ \sum((\text{shifted_scores} - scores) \cdot \epsilon)/h$
end for
traces $\leftarrow$ traces_sum$/n_{\text{iters}}$
$FD \leftarrow \mathbb{E}[\text{traces} + \frac{1}{2} \cdot \|\mathbf{scores}\|^2]$

3. PageBlocks
4. Mammography
5. Magic.gamma
6. Skin
7. Backdoor
8. Glass
9. Lymphography
10. Stamps
11. WDBC
12. SpamBase
13. Hepatites
14. Wine
15. Letter

M. Consistency Theorem

In this Section we state and prove the consistency result for the SOSREP estimators.

Recall that for any $a > 0$, $\mathcal{H}^a$ is the RKHS with the norm given by (9) and the kernel $k^a(x, y)$ given by (11).

For any $f \in \mathcal{H}^1$ define the SOSREP loss

$$L(f) = L^a(f) = \frac{1}{2}\left(-\frac{1}{N}\sum \log f^2(x_i) + \|f\|^2_{\mathcal{H}^a}\right). \tag{108}$$

Note that $f \in \mathcal{H}^1$ if and only if $f \in \mathcal{H}^a$ for every $a > 0$. Recall from the discussion in Section 3.3 and that L is convex when restricted to the open cone

$$\mathcal{C}' = (\mathcal{C}^a)' = \left\{f \in span\left\{k_{x_i}\right\}_1^N \;\middle|\; f(x_i) > 0\right\}. \tag{109}$$

Note that $\mathcal{C}'$ depends on a since the kernel $k_{x_i} = k^a_{x_i}$ depends on a. Observe also that compared to Section 3.3, here we require only positivity on the data points x_i rather than on all $x \in \mathbb{R}^d$, and we restrict the cone to the span of x_i since all the solutions will be obtained there in any case. This of course does not affect the convexity.

The consistency result we prove is as follows:
Theorem 12. Let $x_1, \ldots, x_N$ be i.i.d samples from a compactly supported density v^2 on $\mathbb{R}^d$, such that $v \in \mathcal{H}^1$. Set $a = a(N) = 1/N$, and let $u_N = u(x_1, \ldots, x_N; a(N))$ be the minimizer of the objective (108) in the cone (109). Then $\|u_N - v\|_{L_2}$ converges to 0 in probability.

In words, when N grows, and the regularisation size $a(N)$ decays as $1/N$, the the SOSREP estimators u_N converge to v in L_2.

As discussed in the main text, note also that since $\|v\|_{L_2} = 1$ (as its a density), and since $\|u_N - v\|_{L_2} \to 0$, it follows by the triangle inequality that $\|u_N\|_{L_2} \to 1$. That is, the estimator u_N becomes approximately normalized as N grows.

In Section M.1 below we provide an overview of the proof, and in Section M.2 full details are provided.

M.1. Overview Of The Proof

As discussed in Section 2, consistency for the 1 dimensional case was shown in (Klonias, 1984) and our approach here follows similar general lines. The differences between the arguments are due to the more general multi dimensional setting here, and due to some difference in assumptions.

To simplify the notation in what follows we set $\|\cdot\|_a := \|\cdot\|_{\mathcal{H}^a}$, $\langle\cdot,\cdot\rangle_a := \langle\cdot,\cdot\rangle_{\mathcal{H}^a}$. Recall that $k^a(x, y)$ denotes the kernel corresponding to $\mathcal{H}^a$,

The first step of the proof is essentially a stability result for the optimization of (108), given by Lemma 13 below. In this Lemma we observe that L is strongly convex and thus for any function v, we have

$$\|u - v\|_{\mathcal{H}^a} \leq \|\nabla L(v)\|_{\mathcal{H}^a}, \tag{110}$$

where u is the true minimizer of L in $\mathcal{C}'$ (i.e. the SOSREP estimator). This is particular means that if one can show that the right hand side above is small for the true density v, then the solution u must be close to v. Thus we can concentrate on analyzing the simpler expression $\|\nabla L(v)\|_{\mathcal{H}^a}$ rather than working with the optimizer u directly. Remarkably, this result is a pure Hilbert space result, and it holds for any kernel.

We now thus turn to the task of bounding $\|\nabla L(v)\|_{\mathcal{H}^a}$. As discussed in Section 3.3, the gradient of L in $\mathcal{H}^a$ is given by

$$\nabla L(f) = -\frac{1}{N}\sum f(x_i)^{-1} k_{x_i} + f. \tag{111}$$

Opening the brackets in $\|\nabla L(v)\|^2_{\mathcal{H}^a}$ we have

$$\|\nabla L(v)\|^2_a = \frac{1}{N^2}\sum_{i,j} v^{-1}(x_i)v^{-1}(x_j)k(x_i, x_j) - 2\frac{1}{N}\sum_i v^{-1}(x_i)\langle k_{x_i}, v\rangle_a + \|v\|^2_a \tag{112}$$

$$= \frac{1}{N^2}\sum_{i,j} v^{-1}(x_i)v^{-1}(x_j)k(x_i, x_j) - 2 + \|v\|^2_a. \tag{113}$$

By definitions, we clearly have $\|v\|_a^2 - 1 \to 0$ when $a \to 0$. Thus we have to show that the first term in (113) concentrates around 1. To this end, we will first show that the expectation of

$$\frac{1}{N^2} \sum_{i,j} v^{-1}(x_i) v^{-1}(x_j) k(x_i, x_j) \tag{114}$$

(with respect to x_i's) converges to 1. This is really the heart of the proof, as here we show why v approximately minimizes the SOSREP objective, in expectation, by exploiting the interplay between the kernel, the regularizing coefficient, and the density v. This argument is carried out in Lemmas 14, 16, and 17.

Once the expectation is understood, we simply use the Chebyshev inequality to control the deviation of (114) around its mean. This requires the control of cross products of the terms in (114) and can be achieved by arguments similar to those used to control the expectation. This analysis is carried out in Propositions 18 - 20, and Lemma 21.

One of the technically subtle issues throughout the proof is the presence of the terms $v^{-1}(x_i)$, due to which some higher moments and expectations are infinite. This prevents the use of standard concentration results and requires careful analysis. This is also the reason why obtaining convergence rates and high probability bounds is difficult, although we believe this is possible.

M.2. Full Proof Details

Throughout this section let L_2 be the L_2 space of the the Lebesgue measure on $\mathbb{R}^d$, $L_2 = \left\{ f : \mathbb{R}^d \to \mathbb{C} \;\middle|\; \int_{\mathbb{R}^d} |f|^2 \, dx \leq \infty \right\}$.

Next, we observe that $L(f)$ is *strongly* convex with respect to the norm $\|\cdot\|_{\mathcal{H}^a}$ (see (Nesterov, 2003) for an introduction to strong convexity). As a consequence, we have the following:

Lemma 13. Let u be the minimizer of L in $\mathcal{C}'$. Then for every $v \in \mathcal{C}'$,

$$\|u - v\|_{\mathcal{H}^a} \leq \|\nabla L(v)\|_{\mathcal{H}^a} . \tag{115}$$

Proof. The function $f \mapsto \|f\|_{\mathcal{H}^a}^2$ is 2-strongly convex with respect to $\|\cdot\|_{\mathcal{H}^a}^2$. Since $L(f)$ is obtained by adding a convex function, and multiplying by $\frac{1}{2}$, it follows that $L(f)$ is 1-strongly convex. Strong convexity implies that for all $u, v \in \mathcal{C}'$ we have

$$\langle \nabla L(v) - \nabla L(u), v - u \rangle_a \geq \|v - u\|_a^2 , \tag{116}$$

see (Nesterov, 2003), Theorem 2.1.9. Using the Cauchy Schwartz inequality and the fact that u is a local minimum with $\nabla L(u) = 0$, we obtain

$$\|\nabla L(v)\|_a \cdot \|u - v\|_a \geq \langle \nabla L(v), v - u \rangle_a \geq \|v - u\|_a^2 , \tag{117}$$

yielding the result. □

Now, suppose the samples x_i are generated from a true density $(f^*)^2$. Let $v := f^*$ be the square root of this density. Then, to show that the estimator u is close to v, it is sufficient to show that $\|\nabla L(v)\|_a$ is small. We write $\nabla L(v)$ explicitly:

$$\|\nabla L(v)\|_a^2 = \frac{1}{N^2} \sum_{i,j} v^{-1}(x_i) v^{-1}(x_j) k(x_i, x_j) - 2 \frac{1}{N} \sum_i v^{-1}(x_i) \langle k_{x_i}, v \rangle_a + \|v\|_a^2 \tag{118}$$

$$= \frac{1}{N^2} \sum_{i,j} v^{-1}(x_i) v^{-1}(x_j) k(x_i, x_j) - 2 + \|v\|_a^2 . \tag{119}$$

Since v is a fixed function in $\mathcal{H}^1$, it is clear from definition (9) that $\|v\|_a^2 - 1 \to 0$ as $a \to 0$. Thus, to bound $\|\nabla L(v)\|_a^2$, it suffices to bound

$$\frac{1}{N^2} \sum_{i,j} v^{-1}(x_i) v^{-1}(x_j) k(x_i, x_j) - 1 \tag{120}$$

with high probability over x_i, when N is large and a is small.

Note that the form (11) of the kernel k^a implies that this is a stationary kernel, i.e. $k^a(x,y) = g^a(x-y)$ with

$$g^a(x) = \int_{\mathbb{R}^d} \frac{e^{2\pi i\langle -x,z\rangle}}{1 + a\cdot(2\pi)^{2m}\|z\|^{2m}} dz. \tag{121}$$

Lemma 14. Let $k^a(x,y)$ be the SDO kernel defined by (11). For any function $v \in L_2$, and $x \in \mathbb{R}^d$ set

$$(K_a v)(x) = \int k^a(x,y)v(y)dy. \tag{122}$$

Then K_a is a bounded operator from L_2 to L_2, with $\|K_a\|_{op} \leq 1$ for every $a > 0$. Moreover, for every $v \in L_2$, $\|K_a v - v\|_{L_2} \to 0$ with $a \to 0$.

Proof. Note first that K_a, given by (122), is a convolution operator, i.e. $K_a v = g^a * v = \int g^a(x-y)v(y)dy$. Further, by the Fourier inversion formula, the Fourier transform of g^a satisfies $\mathcal{F}g^a(z) = \frac{1}{1+a\cdot(2\pi)^{2m}\|z\|^{2m}}$ (see Section D.1 for further details), and recall that $\mathcal{F}(g^a * v) = \mathcal{F}g^a \cdot \mathcal{F}v$. Since $|\mathcal{F}g^a(z)| \leq 1$ for every z and $a > 0$, this implies in particular that K_a has operator norm at most 1. Next, by the Plancharel equality we have

$$\begin{aligned}
\|K_a v - v\|_{L_2}^2 &= \|\mathcal{F}(K_a v - v)\|_{L_2}^2 && (123)\\
&= \left\|\mathcal{F}(v)\cdot\left(\frac{1}{1+a\cdot(2\pi)^{2m}\|z\|^{2m}} - 1\right)\right\|_{L_2}^2 && (124)\\
&= \left\|\mathcal{F}(v)\cdot\left(\frac{a\cdot(2\pi)^{2m}\|z\|^{2m}}{1+a\cdot(2\pi)^{2m}\|z\|^{2m}}\right)\right\|_{L_2}^2 && (125)\\
&= \int |\mathcal{F}(v)(z)|^2 \cdot \left(\frac{a\cdot(2\pi)^{2m}\|z\|^{2m}}{1+a\cdot(2\pi)^{2m}\|z\|^{2m}}\right)^2 dz && (126)
\end{aligned}$$

Fix $\varepsilon > 0$. For a radius r denote by $B(r) = \{z \mid \|z\| \leq r\}$ the ball of radius r, and let $B^c(r)$ be its complement. Since $\int |\mathcal{F}(v)(z)|^2 dz < \infty$, there is $r > 0$ large enough such that $\int_{B^c(r)} |\mathcal{F}(v)(z)|^2 dz \leq \varepsilon$. We bound (126) on $B(r)$ and $B^c(r)$ separately. Choose $a > 0$ such that $a \leq \left((2\pi)^{2m}r^{2m}\right)^{-1}\cdot\varepsilon^{\frac{1}{2}}\cdot\|v\|_{L_2}^{-1}$. Then

$$\begin{aligned}
&\int |\mathcal{F}(v)(z)|^2 \cdot \left(\frac{a\cdot(2\pi)^{2m}\|z\|^{2m}}{1+a\cdot(2\pi)^{2m}\|z\|^{2m}}\right)^2 dz && (127)\\
&= \int_{B(r)} |\mathcal{F}(v)(z)|^2 \cdot \left(\frac{a\cdot(2\pi)^{2m}\|z\|^{2m}}{1+a\cdot(2\pi)^{2m}\|z\|^{2m}}\right)^2 dz + \int_{B^c(r)} |\mathcal{F}(v)(z)|^2 \cdot \left(\frac{a\cdot(2\pi)^{2m}\|z\|^{2m}}{1+a\cdot(2\pi)^{2m}\|z\|^{2m}}\right)^2 dz && (128)\\
&\leq \varepsilon\cdot\|v\|_{L_2}^{-1}\int_{B(r)} |\mathcal{F}(v)(z)|^2 dz + \int_{B^c(r)} |\mathcal{F}(v)(z)|^2 dz && (129)\\
&\leq \varepsilon + \varepsilon. && (130)
\end{aligned}$$

This completes the proof. □

Assumption 15. Assume that the density v^2 is compactly supported in $\mathbb{R}^d$, and denote the support by $B = supp(v^2)$.

Note in particular that this implies that B is of finite Lebesgue measure, $\lambda(B) \leq \infty$.

We treat the components with $i \neq j$ and $i = j$ in the sum (120) separately. In the case $i = j$ set $Y_i = v^{-2}(x_i)k(x_i,x_i) = g_a(0)v^{-2}(x_i)$, where g^a was defined in (121). Note that the variable Y_i has an expectation, but does not necessarily have higher moments. Nevertheless, it is still possible to bound the sum using the Marcinkiewicz-Kolmogorov strong law of large numbers, see (Loève, 1977), Section 17.4, point 4°. We record it in the following Lemma, where we also allow a to depend on N, denoting $a = a(N)$, and assume that $a(N)$ does not decay too fast with N.

Lemma 16. Assume that $\lim_{N\to\infty} a^{-\frac{d}{2m}}(N)/N = 0$. Then

$$\frac{g^{a(N)}(0)}{N^2}\sum_i v^{-2}(x_i) \to 0 \tag{131}$$

almost surely, with $N \to \infty$.

Note that since $2m > d$ by construction of the kernels, $a(N) = 1/N$ satisfies the above decay assumption.

Proof. We have

$$\mathbb{E}v^{-2}(x) = \int v^{-2}(x)\cdot v^2(x)\mathbb{1}_{\{B\}}(x)dx = \lambda(B) < \infty. \tag{132}$$

Thus the Marcinkiewicz-Kolmogorov law implies that

$$\frac{1}{N}\sum_i v^{-2}(x_i) \to \lambda(B) \tag{133}$$

almost surely. Next, recall that by Proposition 7, we have $g^a(0) = a^{-\frac{d}{2m}}g^1(0)$. Our decay assumption on $a(N)$ implies then that $g^{a(N)}(0)/N \to 0$, which together with (133) completes the proof. □

Next, for $i \neq j$, set $Y_{ij} = v^{-1}(x_i)v^{-1}(x_j)k(x_i, x_j)$.
Lemma 17. For every $a > 0$ we have $|\mathbb{E}Y_{ij}| \leq 1$, and moreover $\mathbb{E}Y_{ij} \to 1$ when $a \to 0$.

Proof. Recall that v^2 is a density, i.e. $\int v^2(x)dx = 1$, and that the operator K^a was defined in (122). We have

$$\begin{aligned}
|\mathbb{E}Y_{ij} - 1| &= \left|\int v^{-1}(x)v^{-1}(y)k^a(x,y)v^2(x)v^2(y)dxdy - \int v^2(x)dx\right| && (134)\\
&= \left|\int k^a(x,y)v(x)v(y)dxdy - \int v^2(x)dx\right| && (135)\\
&= \left|\int dx\cdot v(x)\left[(K_a v)(x) - v(x)\right]\right| && (136)\\
&\leq \|v\|_{L_2}\|(K_a v) - v\|_{L_2} && (137)\\
&= \|(K_a v) - v\|_{L_2}. && (138)
\end{aligned}$$

The second statement now follows from Lemma 14. For the first statement, write

$$|\mathbb{E}Y_{ij}| = \langle K_a v, v\rangle_{L_2} \leq \|K_a v\|_{L_2}\|v\|_{L_2} \leq 1, \tag{139}$$

where we have used the Cauchy-Schwartz inequality and the first part of Lemma 14. □

It thus remains to establish that $\frac{1}{N^2}\sum_{i\neq j}(Y_{ij} - EY_{ij})$ converges to 0 in probability. We will show this by bounding the second moment,

$$\frac{1}{N^4}\mathbb{E}\left(\sum_{i\neq j}(Y_{ij} - EY_{ij})\right)^2 = \frac{1}{N^4}\sum_{i\neq j, i'\neq j'}(\mathbb{E}Y_{ij}Y_{i'j'} - \mathbb{E}Y_{ij}\mathbb{E}Y_{i'j'}) \tag{140}$$

and then using the Chebyshev inequality. Observe that there are three types of terms of the form $(\mathbb{E}Y_{ij}Y_{i'j'} - \mathbb{E}Y_{ij}\mathbb{E}Y_{i'j'})$ on the right hand side of (140). The first type is when $\{i,j\} = \{i',j'\}$ as sets. Second is when $|\{i,j\}\cap\{i',j'\}| = 1$, and the third is when $\{i,j\}$ and $\{i',j'\}$ are disjoint. In the following three propositions, we bound each type of terms separately.
Proposition 18. There is a function of m, $c(m) > 0$, such that

$$\mathbb{E}Y_{ij}^2 \leq \lambda(B)a^{-\frac{d}{2m}}c(m) < \infty. \tag{141}$$

Proof. Write

$$\mathbb{E}Y_{ij}^2 = \int v^{-2}(x)v^{-2}(y)(k^a(x,y))^2 v^2(x)v^2(y)dxdy \tag{142}$$
$$= \int \mathbb{1}_{\{B\}}(x)\mathbb{1}_{\{B\}}(y)(k^a(x,y))^2 dxdy \tag{143}$$
$$\leq \int \mathbb{1}_{\{B\}}(x)\left\|k_x^a\right\|_{L_2}^2 dx \tag{144}$$
$$= \lambda(B)a^{-\frac{d}{2m}}c(m), \tag{145}$$

where we have used Lemma 8 to compute $\left\|k_x^a\right\|_{L_2}^2$. □

For the $|\{i,j\}\cap\{i',j'\}| = 1$ case we have

Proposition 19. Let i, j, t be three distinct indices. Then

$$|\mathbb{E}Y_{ij}Y_{jt}| \leq 1. \tag{146}$$

Proof.

$$\mathbb{E}Y_{ij}Y_{jt} = \int v^{-1}(x_i)v^{-1}(x_j)k^a(x_i,x_j)v^{-1}(x_j)v^{-1}(x_t)k^a(x_j,x_t)v^2(x_i)v^2(x_j)v^2(x_t)dx_idx_jdx_t \tag{147}$$
$$= \int v(x_i)k^a(x_i,x_j)v(x_t)k^a(x_t,x_j)\mathbb{1}_{\{B\}}(x_j)dx_idx_jdx_t \tag{148}$$
$$= \int (K_av)^2(x_j)\mathbb{1}_{\{B\}}(x_j)dx_j \tag{149}$$
$$\leq \int (K_av)^2(x_j)dx_j \tag{150}$$
$$= \left\|K_av\right\|_{L_2}^2 \tag{151}$$
$$\leq 1, \tag{152}$$

where we have used Lemma 14 on the last line. □

Finally, for the disjoint case,

Proposition 20. Let i, j, i', j' be four distinct indices. Then

$$\mathbb{E}(Y_{ij} - EY_{ij})(Y_{i'j'} - EY_{i'j'}) = 0. \tag{153}$$

Proof. When i, j, i', j' are distinct, Y_{ij} and $Y_{i'j'}$ are independent. □

We now collect these results to bound (140).

Lemma 21. There is a function $c'(m,B) > 0$ of m, B, such that, choosing $a(N) = 1/N$, we have

$$\frac{1}{N^4}\mathbb{E}\left(\sum_{i\neq j}(Y_{ij} - EY_{ij})\right)^2 \leq \frac{c'(m,B)}{N} \tag{154}$$

for every $N > 0$.

Proof. Observe first that all the expressions of the form $\mathbb{E}Y_{ij}\mathbb{E}Y_{i'j'}$ are bounded by 1, by Lemma 17. Next, note that there are $O(N^2)$ terms of the first type. Since $2m > d$, we have $a^{-\frac{d}{2m}} = a^{-\frac{d}{2m}}(N) \leq N$, and thus, the overall contribution of such terms to the sum in (154) is $O(N^{-4}\cdot N\cdot N^2) = O(1/N)$. Similarly, there are $O(N^3)$ terms of the second type, each bounded by constant, and thus the overall contribution is $O(N^{-4}\cdot N^3) = O(1/N)$. And finally, the contribution of the terms of the third type is 0. □

We now prove the main consistency Theorem.

Of Theorem 3. First, observe that by definition $\|u - v\|_{L_2} \leq \|u - v\|_a$ for any $u, v \in \mathcal{H}^1$. Next, by Lemma 13 and using (119), we have

$$\|u_N - v\|_{L_2} \leq \|u_N - v\|_a \leq \frac{1}{N^2} \sum_{i,j} v^{-1}(x_i) v^{-1}(x_j) k(x_i, x_j) - 2 + \|v\|_a^2 . \tag{155}$$

Clearly, by definition (9), we have $\|v\|_a^2 - 1 \to 0$ as $a \to 0$. Write

$$\frac{1}{N^2} \sum_{i,j} v^{-1}(x_i) v^{-1}(x_j) k(x_i, x_j) - 1 = \frac{1}{N^2} \sum_i Y_i + \frac{1}{N^2} \sum_{i \neq j} (Y_{ij} - \mathbb{E} Y_{ij}) + \left[\frac{N^2 - N}{N^2} \cdot \mathbb{E} Y_{12} - 1 \right]. \tag{156}$$

The last term on the right hand side converges to 0 deterministically with a and N, by Lemma 17. The first terms converges strongly, and hence in probability, to 0, by Lemma 16. And finally, the middle term converges in probability to 0 by Lemma 21 and by Chebyshev inequality. □

N. Comparison of SOSREP for Different Kernels

In this section, we evaluate the SOSREP density estimator with various kernels on the ADBench task. Specifically, we consider the SDO kernel given by (11), the L_2 Laplacian kernel, and the Gaussain kernel. The Laplacian (aka Exponential) kernel is given by

$$k(x, y) = (1/\sigma^d) \cdot e^{-\|x-y\|/\sigma} \tag{157}$$

for $\sigma > 0$, where $\|x - y\|$ is the Euclidean norm on $\mathbb{R}^d$. The Gaussian kernel is given by

$$k(x, y) = (1/\sigma^d) \cdot e^{-\|x-y\|^2/(2\sigma^2)}. \tag{158}$$

Generally, in this task, we observe a similar behavior across all kernels, although the Gaussian kernel underperforms on a few datasets.

Figure 17 presents the performance of each kernel on the ADBench benchmark (see section 6.1). Specifically, we plot the test set AUC values on the Y-axis against the different datasets on the X-axis. Note that due to convexity, the results are nearly identical for different initializations. Figures 16 and 15 display the ADbench results in a manner consistent with Section 6.1. It is noteworthy that the Laplacian kernel also achieves impressive results. Furthermore, when considering ranks and thus mitigating the impact of extreme AUC values, the Gaussian kernel demonstrates significantly improved performance.

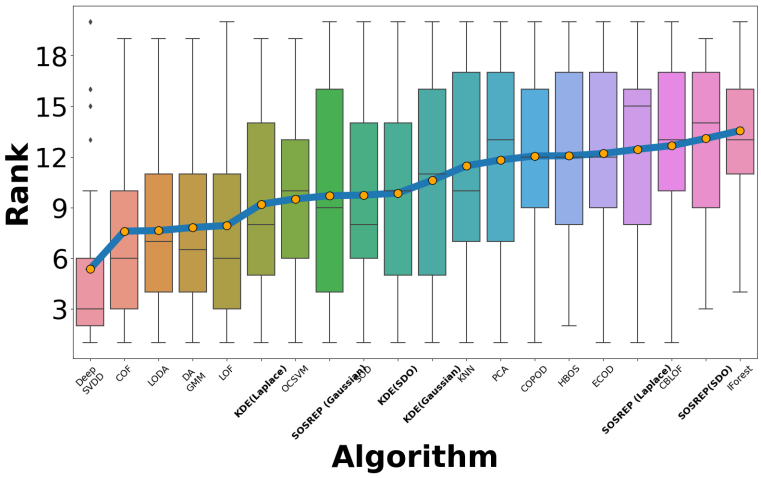

Figure 15: Anomaly Detection Results on ADBench. Mean Relative Ranking Per Dataset, Higher is Better.

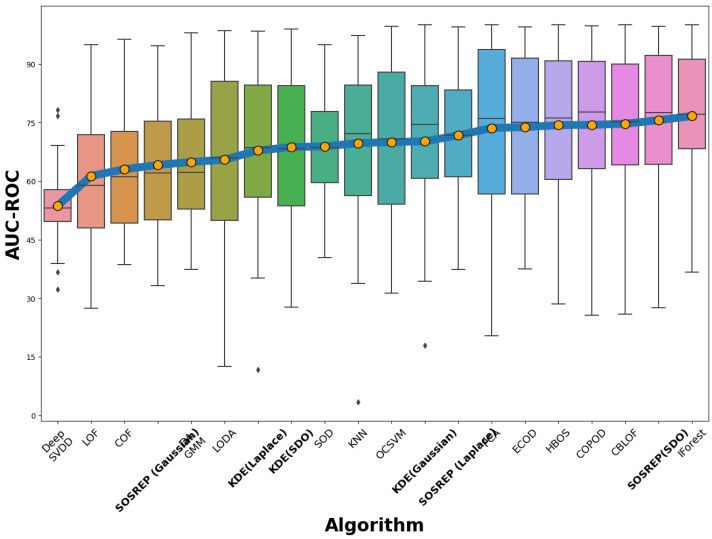

Figure 16: Anomaly Detection Results on ADBench. Mean AUC Per Dataset, Higher is Better.

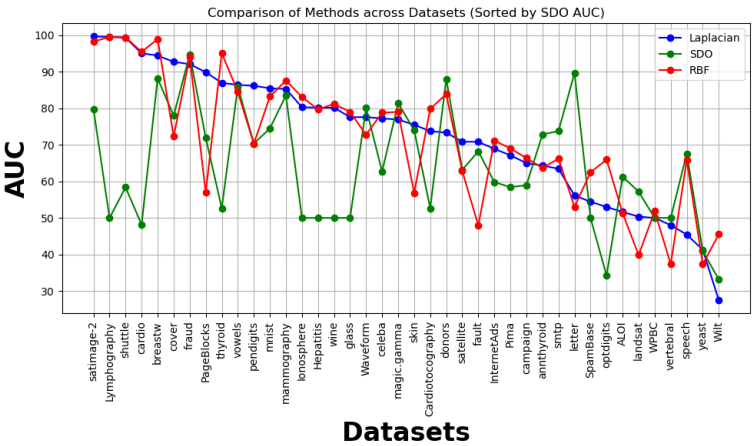

Figure 17: Comparing SOSREP results on ADBench for different kernels.

Figure 18: Evaluating SOSREP with various kernels on the ADbench benchmark.

O. Full Table of Contents for the Supplementary materials

1. **Basic Minimizer Properties**
   - Discussion on the minimizer of the SOSREP objective and its properties.
   - Proofs for properties of the minimizer.
   - Lemma on minimization properties.
   - Detailed proofs for the properties of the minimizer.
2. **Derivation of the Gradients, Proof of Lemma**
   - Derivation of the standard and natural gradients of the objective.
   - Proof of the lemma regarding gradients.
3. **SDO Kernel Details**
   - Full proof of Theorem on kernel form.
   - Additional details on sampling approximation of SDO.
   - SDO Kernel Derivation.
   - Sampling Approximation details.
4. **A Few Basic Properties of the Kernel**
   - Propositions on the real-valued nature of the kernel and its properties.
   - Proof of properties related to the kernel.
5. **Difference between SOSREP and KDE Models**
   - Analysis of the differences between SOSREP and KDE estimators.
   - Empirical evaluation of the difference for real data.
6. **KDE vs SOSREP Comparison Proofs**
   - Development of proofs for comparing KDE and SOSREP.
   - Solution of SOSREP for a 2-Block Model.
   - Proof of Proposition.
7. **Invariance of $\mathcal{C}$ under Natural Gradient**
   - Discussion on the invariance of the non-negative cone under natural gradient steps.
   - Proposition on invariance and its proof.
8. **Fisher Divergence for Hyper-Parameters Selection**
   - The rationale behind using Fisher Divergence for hyperparameter selection.
   - Score Matching and Fisher Divergence details.
   - The Hessian Estimation for Small a's.
   - Approximating the Hessian Trace.
9. **Experiments**
   - Detailed analysis of AUC-ROC performance.
   - Discussion on Duplicate Anomalies and its impact.
   - Hyperparameter Tuning with On-the-fly Fisher-Divergence Minimization.
10. **Algorithm Overview**
   - Detailed overview and breakdown of the main algorithm used in the paper.
   - Specific algorithms for sampling the multidimensional SDO Kernel, calculating optimal alphas, density over coordinates, Fisher Divergence with Hessian Trace Approximation.

11. **Figure 3 - List of Datasets**
    - Listing of datasets used for comparing fractions of negative values for alpha optimization vs natural gradients.
12. **Consistency Theorem**
    - Statement and proof of the consistency result for SOSREP estimators.
    - Detailed proof and related lemmas.
13. **Comparison of SOSREP for Different Kernels**
    - Evaluation of SOSREP with various kernels on the ADBench task.
    - Performance comparison and discussion.