Unnormalized Density Estimation with Root Sobolev Norm Regularization

Mark Kozdoba
Binyamin Perets
Shie Mannor
Technion – Israel Institute of Technology.

Abstract

Density estimation is one of the most central problems in statistical learning. In this paper we introduce a new approach to non-parametric density estimation that is statistically consistent, is provably different from Kernel Density Estimation, makes the inductive bias of the model clear and interpretable, and performs well on a variety of relatively high dimensional problems.

One of the key points of interest in terms of optimization is our use of natural gradients (in Hilbert Spaces). The optimization problem we solve is non-convex, and standard gradient methods do not perform well. However, we show that the problem is convex on a certain positive cone, and natural gradient steps preserve this cone. The standard gradient steps, on the other hand, tend to lose positivity. This is one of the few cases in the literature where the reasons for the practical preference for the natural gradient are clear.

In more detail, our approach is based on regularizing a version of a Sobolev norm of the density, and there are several core components that enable the method. First, while there is no closed analytic form for the associated kernel, we show that one can approximate it using sampling. Second, appropriate initialization and natural gradients are used as discussed above. Finally, while the approach produces unnormalized densities, which prevents the use of cross-validation, we show that one can instead adopt the Fisher Divergence-based Score Matching methods for this task. We evaluate the resulting method on a comprehensive recent tabular anomaly detection benchmark suite which contains more than 15 healthcare and biology-oriented data sets (ADBench), and find that it ranks second best, among more than 15 algorithms.

1. Introduction

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 [29], Neural ODEs [6], and Score Based methods, [38]. However, the situation appears to be different for non parametric density estimation methods, [43], [15]. While there is recent work for low dimensional (one or two dimensional) data, see for instance [40], [41], [7], [9] and survey [20], there still are very few non-parametric methods applicable in higher dimensions.

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. In this paper we introduce and study a density estimator of the following form:

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

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

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

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 non-negative, and $\|f\|_{\mathcal{H}^a} < \infty$ implies $\int_{\mathbb{R}^d}(f^*)^2(x)dx < \infty$. Thus $(f^*)^2$ is integrable, although not necessarily integrates to 1. 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 the derivatives of f^* of certain order. 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 norm L_2 norm of f^* and maintain computabilty, since L_2 is not an RKHS. However, adding the derivatives with any $a > 0$ makes the space into an RKHS which allows to control smoothness, hence, we call the objective Root Sobolev Regularized density estimator (RSR).

Despite it being natural and simple, the objective (1) has not been studied in the literature as a tool for multidimensional data analysis. It has been introduced and studied in [10] and [21]; see also [8], but solely on 1d case. Our goal in this paper is to develop the necessary ingredients for RSR to become useful in high dimensions.

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. However, we show that it can be approximated by a sampling procedure.

Next, standard gradient descent optimization of (1) produces poor results. We show that this can be solved by using *natural gradients* ([2], [25]). The concept of a gradient that is independent of a parametrisation was proposed in [2], and in [25]. In [18] it was introduced into Reinforcement Learning, where it is widely used today. Here we consider specifically a Hilbert Space version of the notion, which also has a variety of applications, although typically not in RL. See for instance [25], [45], and [34] for a sample of early and more recent applications. Natural Gradient in Hilbert Spaces is also referred to as Functional Gradient in the literature. While we are not aware of a dedicated treatment of the subject, introductory notes may be found at [3] and in the works cited above.

Solutions of (1) are unnormalized, complicating hyperparameter tuning by precluding the use of maximum likelihood. We address this by leveraging score-matching [17, 36, 37] to assess the Fisher Divergence, a technique that has recently garnered renewed interest. We also offer examples proving that RSR can markedly differ from the standard Kernel Density Estimator (using the same kernel). Hence, RSR is a genuinely new estimator, with different properties than KDE (see Section J).

Finally, consistency of the RSR in one dimension was shown in [21]. In Appendix Section O we state and prove the consistency of the RSR in any fixed dimension d, for compactly supported ground truth densities. Note that as a result of consistency, it can also be shown that the estimator $(f^*)^2$ becomes normalized, at least asymptotically with N. Note that applications like anomaly detection and standard MCMC sampling algorithms, such as Langevin Dynamics or Hamiltonian Monte Carlo, do not require normalization. With these contributions in place, we show that RSR achieves the remarkable scoring *second best* on a recent comprehensive Anomaly Detection benchmark, [14], which includes more than 45 datasets and more than 15 specialized AD methods.

See Supplementary Material Section A for a full summary of the literature and related work.

2. The RSR Desnity Estimator

This Section describes the general RSR Framework, formulated in an abstract Reproducing Kernel Hilbert Space. We first introduce the general optimization problem and discuss a few of its properties. In Section D we discuss the gradient descent optimization and introduce the natural gradient.

2.1. The Basic Framework

Let $\mathcal{X}$ be a set and let $\mathcal{H}$ be a Reproducing Kernel Hilbert Space (RKHS, [33]) 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 $\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}}$, and the subscript $\mathcal{H}$ may be dropped when it is clear from context. Given a set of points $S = \{x_1, \ldots, x_N\} \subset \mathcal{X}$, we define the RSR estimator as the solution to the following optimization problem:

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

For appropriate spaces $\mathcal{H}$, the function $(f^*)^2$ corresponds to an unnormalized density (note that $\int_{\mathbb{R}^d} (f^*)^2(x)dx < \infty$). By the Representer Theorem for RKHS, the minimizer of (3) has the form

$$f(x) = f_\alpha(x) = \sum_{i=1}^{N} \alpha_i k_{x_i}(x), \text{ for some } \alpha = (\alpha_1, \ldots, \alpha_N) \in \mathbb{R}^N . \tag{4}$$

Thus one can solve (3) by optimizing over a finite dimensional vector α. Notice that standard RKHS problems typically use $\lambda \|h\|_{\mathcal{H}}^2$, where $\lambda > 0$ controls the regularization. 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. In addition, any solution of (3) satisfies $\|f\|_{\mathcal{H}}^2 = 1$. See Lemma 5 in Appendix for full details on these two points.

3. Non-Convex Optimization

Observe that the objective

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

is not convex in f 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 as the restriction of L the positive cone of functions $\mathcal{C} = \{f \mid f(x) \geq 0 \ \forall x \in \mathcal{X}\}$. 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 restrict the optimization to the positive cone $\mathcal{C}$. 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 4.1) 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*, as discussed 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 K). This does not seem to be true for the regular gradient. Empirically, this results in a more stable algorithm and further performance improvement. A comparison of standard and natural gradients w.r.t negative values is given in Section E.

3.1. Gradients and Minimization

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 (18).

As detailed in 2.1, 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, [26], 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*

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

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: (λ is the learning rate)

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

3.2. Natural-Gradient vs Standard Gradient Comparison

We conduct an experiment to demonstrate that the standard gradient descent may significantly amplify the fraction of negative values in a solution, while the natural gradient keeps it constant. See also the discussion in Section 2.1. We have randomly chosen 15 datasets from ADBench, and for each dataset we have used 50 non negative α initializations. Then we have run both algorithms for 1000 iteartions.

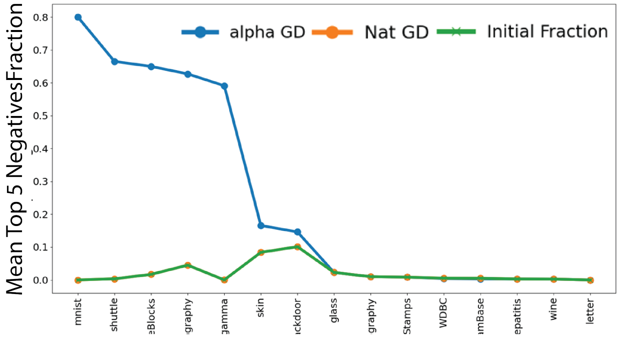

Figure 1: Fraction of negative values for natural versus α gradient-based optimization across datasets. The X-axis represents datasets from ADbench (see Supplementary Material Section P for details).

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 highest 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, blue) yields about 70% negative values in the 5 worst cases (i.e. 10% of initializations).

4. Single Derivative Order Kernel Approximation

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

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 $\|f\|_{L_2}^2 = \int |f(x)|^2 \, dx$. Then, for $m \in \mathbb{N}$ and $a > 0$ denote

$$\|f\|_a^2 = \|f\|_{L_2}^2 + a \sum_{|\kappa|_1 = m} \frac{m!}{\kappa!} \|(D^\kappa f)\|_{L_2}^2 . \tag{8}$$

The norm $\|f\|_a^2$ induces a topology that is equivalent to that of a standard L_2 Sobolev space ([1], [32]) of order m. 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 $\mathcal{H}^a = \left\{ f : \mathbb{R}^d \to \mathbb{C} \;\middle|\; \|f\|_a^2 < \infty \right\}$ be the space of functions with a finite $\|f\|_a^2$ norm. Denote by

$$\langle f, g \rangle_{\mathcal{H}^a} = \langle f, g \rangle_{L_2} + a \sum_{|\alpha|_1 = m} \frac{m!}{\kappa!} \langle (D^\kappa f), (D^\kappa g) \rangle_{L_2}^2 \tag{9}$$

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 $\langle k^a_x, f\rangle_{\mathcal{H}^a} = f(x)$ for all $f \in \mathcal{H}^a$ and $x \in \mathbb{R}^d$. The kernel is given by*

$$k^a(x,y) = \int_{\mathbb{R}^d} \frac{e^{2\pi i\langle y-x,z\rangle}}{1 + a\cdot(2\pi)^{2m}\left\|z\right\|^{2m}}dz = \int_{\mathbb{R}^d} \frac{1}{1 + a\cdot(2\pi)^{2m}\left\|z\right\|^{2m}} \cdot e^{2\pi i\langle y,z\rangle} \cdot \overline{e^{2\pi i\langle x,z\rangle}}dz. \quad (10)$$

The proof of Theorem 2 follows the standard approach of deriving kernels in Sobolev spaces, via computation and inversion of the Fourier transform ([32]). However, compact expressions such as (10) are only possible for some choices of derivative coefficients. Since the form (8) was not previously considered in the literature, we provide the full proof in the Appendix.

4.1. 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. However, for $d > 1$, it seems unlikely that there are closed expressions (see [28]). To resolve this, note that (10) may be interpreted as an average of $e^{2\pi i\langle y,z\rangle} \cdot \overline{e^{2\pi i\langle x,z\rangle}}$, where z is sampled from an unnormalized density $w^a(z) = (1 + a\cdot(2\pi)^{2m}\left\|z\right\|^{2m})^{-1}$ on $\mathbb{R}^d$. This 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 [30]. There, it was observed that by Bochners's Theorem, [31], 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

$$\hat{k}^a(x,y) = \frac{1}{T}\sum_{t=1}^{T} \cos\left(\langle z_t, x\rangle + b_t\right) \cdot \cos\left(\langle z_t, y\rangle + b_t\right), \quad (11)$$

where b_t are additional i.i.d samples, sampled uniformly from $[0, 2\pi]$. 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).

For the case of interest in this paper, the SDO kernel, Bochner's representation is given by (10) in Theorem 2. Thus, to implement the sampling scheme (11) it remains to describe how one can sample from $w^a(z)$ 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 Appendix for full details on this change of variables). Next, note that sampling θ is easy ([24]), 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 sufficiently fine grained discretization of the line was sufficient.

5. Experiments

We evaluate our method on real-world unsupervised Anomaly Detection (AD) tasks, where normalized density is not a concern. AD's inherent traits, such as differentiating latent from out-of-distribution samples, make it apt for this assessment. We compare our results to a gold standard AD benchmark, ADbench ([14]), which evaluates a broad array of 15 AD algorithms across more than 47 labeled datasets, primarily targeting healthcare and natural science-oriented tabular data. In addition, we evaluate KDE using both Gaussian and Laplace kernels, and as an ablation study, we compare RSR to KDE with SDO kernel. In addition to AUC-ROC, we also focus on the relative ranking in each dataset. For both AUC-ROC and rank evaluations, **RSR emerges as the 2nd best AD method overall**. Notably, this achievement is with the 'vanilla' version of our method, without any pre or post-processing dedicated to AD. In contrast, many other methods are specifically tailored for AD and include extensive pre and post-processing. In addition to performing well on the standard ADBench benchmark, and perhaps even more impressively, RSR excels also on the more demanding setup of *duplicate anomalies*, which was also extensively discussed in [14]. Here, **RSR rises to the forefront as the top AD method** (with a lead of 4% over the closest contender). This scenario, analogous to many practical situations, especially in biological data where "missing data" appears as an unknown constant, results in significant performance drops for previous front-runners like Isolation Forest. See Appendix M for the associated figures, full details and additional results.

6. Acknowledgments

This Project has received funding from the European Union's Horizon Europe Programme under grant agreement No.101084642.

References

[1] Robert A Adams and John JF Fournier. *Sobolev spaces*. Elsevier, 2003.

[2] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2): 251–276, 1998.

[3] Drew Bagnell. Functional gradient descent, 2012. https://www.cs.cmu.edu/~16831-f12/notes/F12/16831_lecture21_danielsm.pdf,.

[4] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[5] Guilherme O. Campos, Arthur Zimek, Jörg Sander, Ricardo J. Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E. Houle. On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Min. Knowl. Discov.*, 30(4): 891–927, jul 2016. ISSN 1384-5810. doi: 10.1007/s10618-015-0444-8. URL https://doi.org/10.1007/s10618-015-0444-8.

[6] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[7] Zhenyu Cui, Justin Lars Kirkby, and Duy Nguyen. Nonparametric density estimation by b-spline duality. *Econometric Theory*, 36(2):250–291, 2020.

[8] Paulus Petrus Bernardus Eggermont, Vincent N LaRiccia, and VN LaRiccia. *Maximum penalized likelihood estimation. Volume I: Density Estimation*, volume 1. Springer, 2001.

[9] Federico Ferraccioli, Eleonora Arnone, Livio Finos, James O Ramsay, Laura M Sangalli, et al. Nonparametric density estimation over complicated domains. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY SERIES B STATISTICAL METHODOLOGY*, 83(2):346–368, 2021.

[10] IJ Good and Ray A Gaskins. Nonparametric roughness penalties for probability densities. *Biometrika*, 58(2):255–277, 1971.

[11] Loukas Grafakos. *Classical fourier analysis*, volume 2. Springer, 2008.

[12] Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[13] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. 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, 2016.

[14] Songqiao Han, Xiyang Hu, Hailiang Huang, Mingqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark, 2022.

[15] Wolfgang Härdle, Marlene Müller, Stefan Sperlich, Axel Werwatz, et al. *Nonparametric and semiparametric models*, volume 1. Springer, 2004.

[16] M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18(3):1059–1076, 1989. doi: 10.1080/03610918908812806. URL https://doi.org/10.1080/03610918908812806.

[17] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[18] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

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

[20] J Lars Kirkby, Álvaro Leitao, and Duy Nguyen. Spline local basis methods for nonparametric density estimation. *Statistic Surveys*, 17:75–118, 2023.

[21] VK Klonias. On a class of nonparametric density and regression estimators. *The Annals of Statistics*, pages 1263–1284, 1984.

[22] Donghwoon Kwon, Kathiravan Natarajan, Sang C. Suh, Hyunjoo Kim, and Jinoh Kim. 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, 2018. doi: 10.1109/ICDCS.2018.00178.

[23] M Loève. *Probability theory I*. Springer, 1977.

[24] George Marsaglia. Choosing a Point from the Surface of a Sphere. *The Annals of Mathematical Statistics*, 43(2):645 – 646, 1972. doi: 10.1214/aoms/1177692644. URL https://doi.org/10.1214/aoms/1177692644.

[25] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent. *Advances in neural information processing systems*, 12, 1999.

[26] James R Munkres. *Analysis on manifolds*. CRC Press, 2018.

[27] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[28] Erich Novak, Mario Ullrich, Henryk Woźniakowski, and Shun Zhang. Reproducing kernels of sobolev spaces on rd and applications to embedding constants and tractability. *Analysis and Applications*, 16(05):693–715, 2018.

[29] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

[30] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[31] Walter Rudin. *Fourier analysis on groups*. Courier Dover Publications, 2017.

[32] Saburou Saitoh and Yoshihiro Sawano. *Theory of reproducing kernels and applications*. Springer, 2016.

[33] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[34] Zebang Shen, Zhenfu Wang, Alejandro Ribeiro, and Hamed Hassani. Sinkhorn barycenter via functional gradient descent. *Advances in Neural Information Processing Systems*, 33:986–996, 2020.

[35] Shashank Singh, Ananya Uppal, Boyue Li, Chun-Liang Li, Manzil Zaheer, and Barnabás Póczos. Nonparametric density estimation under adversarial losses. *Advances in Neural Information Processing Systems*, 31, 2018.

[36] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

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

[38] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[39] Elias M Stein and Guido Weiss. 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, 1971.

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

[41] Ananya Uppal, Shashank Singh, and Barnabás Póczos. Nonparametric density estimation & convergence rates for gans under besov ipm losses. *Advances in neural information processing systems*, 32, 2019.

[42] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

[43] Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[44] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

[45] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Appendix A. Literature and Related Work

A as discussed in Section 1, a scheme that is equivalent to (1) was studied in [10] and [21]; see also [8]. However, these works concentrated solely on 1d case, and used spline methods to solve (3) in the special case that amounts to the use of one particular kernel. Our more general RKHS formulation in Section (2.1) allows the use of a variety of kernels. Most importantly, however, as discussed in Section (1), in this work we have developed and evaluated the high dimensional version of RSR.

The most common non parametric density estimator is the Kernel Density Estimator (KDE), [15, 43]. For comparison, we have evaluated KDE, with the two most popular kernels, Gaussian and Laplacian, on the AD benchmark. However, these methods did not perform well (Section 5) on this task. We have also evaluated KDE with the SDO kernel that we introduce in Section 4, and which has not been previously considered in the literature for $d > 1$. Remarkably, we find that using this kernel significantly improves the AD performance compared to Gaussain and Laplacian kernels. However, the performance is still subpar to the RSR estimator.

Another common group of non parametric density estimators are the *projection methods*, [42]. These methods have mostly been studied in one dimensional setting, see the survey [20]. It is worth noting that with the exception of [41], 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 [35], extending a classical work [19]. However, to the best of our knowledge, such methods have never have been practically applied in high dimensions.

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. The divergence between data and a model can be approximated via the methods of [17], which have been recently computationally improved in [37] in the context of score based generative models, [36]. Here we use the FD as a quality metric for hyperparameter selection. In particular, we adapt the Hutchinson trace representation based methods used in [37] and [12] to the case of models of the form (3). Full details are given in the Appendix Section L.

The concept of a gradient that is independent of a parametrisation was proposed in [2], and in [25]. In [18] it was introduced into Reinforcement Learning, where it is widely used today. Here we consider specifically a Hilbert Space version of the notion, which also has a variety of applications, although typically not in RL. See for instance [25], [45], and [34] for a sample of early and more recent applications. Natural Gradient in Hilbert Spaces is also referred to as Functional Gradient in the literature. While we are not aware of a dedicated treatment of the subject, introductory notes may be found at [3] and in the works cited above.

Consistency of the RSR in one dimension was shown in [21], for kernels that coincide with SDO in one dimension. In Appendix Section O we state and prove the consistency of the RSR in any fixed dimension d, for compactly supported ground truth densities. Our approach generally follows the same lines as that of [21]. However, some of the estimates are done differently, since the corresponding arguments in [21] were intrinsically one dimensional.

Appendix B. Difference between RSR and KDE Models

In this Section we construct an analytic example where the RSR 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

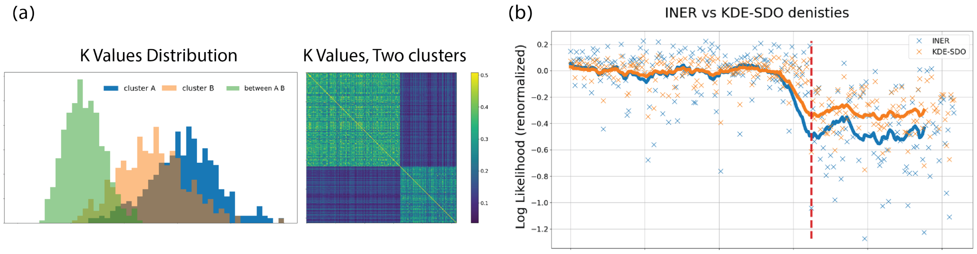

Figure 2: (a) Distribution of Kernel Values and SDO Kernel Values Inside and Between Clusters. (b) RSR and KDE Loglikelihoods. the x-axis represents points in the data, arranged by clusters, y-axis shows the log-likelihood.

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 RSR, 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 RSR 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

$$f_{kde}(x) = f_{kde,S}(x) = \frac{1}{|S|} \sum_i k_{x_i}(x). \tag{12}$$

Let f_{RSR} be the solution of (3). We will compare the ratios $f_{kde}(x_i)/f_{kde}(x_j)$ versus the corresponding quantities for RSR, $f^2_{RSR}(x_i)/f^2_{RSR}(x_j)$ for some pairs x_i, x_j. Note that these ratios do not depend on the normalization of f_{kde} and f^2_{RSR}, and can be computed from the unnormalized versions. In particular, we do not require k_{x_i} to be normalized in (12).

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

$$K = \begin{cases} k(x_i, x_i) = k(x'_j, x'_j) = 1 & \text{for all } i \leq N, j \leq M \\ k(x_i, x_j) = \gamma^2 & \text{for } i \neq j \\ k(x'_i, x'_j) = \gamma'^2 & \text{for } i \neq j \\ k(x_i, x'_j) = \beta\gamma\gamma' & \text{for all } i, j \end{cases} \tag{13}$$

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 (13) often appear in real data. See Section B.1 for an illustration. In particular, Figure 2 show a two cluster configuration in that data, and the distribution of $k(x, x')$ values.

The KDE estimator for K is simply

$$f_{kde}(x_t) = \frac{1}{N+M}\left[1 + (N-1)\gamma^2 + M\beta\gamma\gamma'\right] \approx \frac{N}{N+M}\gamma^2 + \frac{M}{N+M}\beta\gamma\gamma', \tag{14}$$

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

$$\frac{f_{kde}(x_t)}{f_{kde}(x'_t)} = \frac{\gamma^2 + \beta\gamma\gamma'}{\gamma'^2 + \beta\gamma\gamma'}. \tag{15}$$

The derivation of the RSR estimator is considerably more involved. Here we sketch the argument, while full details are given in Appendix Section J. 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 $\beta = (a, \ldots, a, b, \ldots, b)$, 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:

$$\begin{cases} a & = a^{-1}\left[1 + (N-1)\gamma^2\right] + b^{-1}M\beta\gamma\gamma' \\ b & = a^{-1}N\beta\gamma\gamma' + b^{-1}\left[1 + (M-1)\gamma'^2\right] \end{cases} \tag{16}$$

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 3 *Consider the kernel and point configuration described by* (13)*, with $M = N$. Then for every $x_t \in S_1, x'_s \in S_2$,*

$$\frac{f_{RSR}(x_t)}{f_{RSR}(x'_s)} = \frac{\gamma^2}{\gamma'^2}. \tag{17}$$

In particular, the ratio does not depend on β.

It remains to compare the ratio (17) to KDE's ratio (15). 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 (15) 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 RSR there is no such influence, and the ratio (17) may be arbitrarily large. We thus expect the gap between the cluster densities to be larger for RSR, which is indeed the case empirically. One occurence of this on real data is illustrated in Figure 2.

B.1. Evaluation of the Difference Between RSR and KDE for Real Data

We have performed spectral clustering of the "letter" dataset from ADBech ([14]), using the empirical SDO kernel as affinity matrix for both RSR and KDE. We then have chosen two clusters that most resemble the two block model (13) in Section B. The kernel values inside and between the clusters are shown in Figure 2a. Next, we train the RSR and KDE models for just these two clusters (to be compatible with the setting of Section B. The results are similar for densities trained on full data). The log of these RSR and KDE densities in shown in Figure 2b (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 RSR model, yielding a less uniform model, as expected from the theory.

Appendix C. Fisher-Divergence Based Hyperparameter Tuning

As noted in Section 1, dealing with unnormalized densities adds a layer of complexity to hyperparameter tuning since it prevents the use of maximum likelihood for determining optimal parameters. Consequently, for tuning the smoothness parameter $a > 0$ in both RSR and KDE with SDO kernel we employ a score-based approach. This approach measures the Fisher Divergence (FD) between a dataset sampled from an unknown distribution and a proposed distribution. Specifically, in our case, the FD between the density learned on the training set and the density inferred on the test set. Hence, the hyperparameters tuning procedure simply picks the parameters that resolve with the lowest FD. A thorough explanation of this approach, addressing important computational aspects and featuring a dedicated algorithm figure, can be found in Appendix Section L.

Appendix D. Gradients and Minimization

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 (18).

As detailed in 2.1, 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, [26], 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 4 (Gradients) *The standard and the natural gradients of $L(f)$ are given by*

$$\nabla_\alpha L = 2\left[K\alpha - \frac{1}{N} K (K\alpha)^{-1}\right] \in \mathbb{R}^N \textit{ and } \nabla_f L = 2\left[f - \frac{1}{N}\sum_{i=1}^{N} f^{-1}(x_i) k_{x_i}\right] \in \mathcal{H} \qquad (18)$$

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:

$$\alpha \leftarrow \alpha - 2\lambda \left[K\alpha - \frac{1}{N} K (K\alpha)^{-1}\right] \text{ and } \alpha \leftarrow \alpha - 2\lambda \left[\alpha - \frac{1}{N} (K\alpha)^{-1}\right], \qquad (19)$$

where λ is the learning rate.

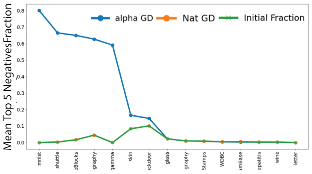

Figure 3: Fraction of negative values for natural versus α gradient-based optimization across datasets. The X-axis represents datasets from ADbench (see Appendix Section P for details).

Appendix E. Natural-Gradient vs Standard Gradient Comparison

We conduct an experiment to demonstrate that the standard gradient descent may significantly amplify the fraction of negative values in a solution, while the natural gradient keeps it constant. See also the discussion in Section 2.1. We have randomly chosen 15 datasets from ADBench, and for each dataset we have used 50 non negative α initializations. Then we have run both algorithms for 1000 iteartions. 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 highest 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, blue) yields about 70% negative values in the 5 worst cases (i.e. 10% of initializations).

Appendix F. Basic Minimizer Properties

As discussed in Section 2.1, the minimizer of the RSR 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 5 *Define*

$$f = \operatorname*{argmin}_{h \in \mathcal{H}} -\frac{1}{N} \sum_i \log h^2(x_i) + \|h\|_{\mathcal{H}}^2 . \tag{20}$$

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

$$f' = \operatorname*{argmin}_{h \in \mathcal{H}} -\frac{1}{N} \sum_i \log h^2(x_i) + \lambda^2 \|h\|_{\mathcal{H}}^2 , \tag{21}$$

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

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

$$\operatorname*{argmin}_{a>0} -\frac{1}{N} \sum_i \log(ah)^2(x_i) + \|ah\|^2 = \tag{22}$$

$$\operatorname*{argmin}_{a} -\frac{1}{N} \sum_i \log h^2(x_i) - \log a^2 + a^2 \|h\|^2 . \tag{23}$$

Taking derivative w.r.t a we have

$$-\frac{2a}{a^2} + 2a \|h\|^2 = 0. \tag{24}$$

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 (22) to be the minimizer in (20), $h = f$. 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 $g = \lambda h$,

$$\begin{aligned}
&\operatorname*{argmin}_{h\in\mathcal{H}} -\frac{1}{N}\sum_i \log h^2(x_i) + \lambda^2 \|h\|^2 && (25)\\
&= \lambda^{-1} \operatorname*{argmin}_{g\in\lambda\mathcal{H}=\mathcal{H}} -\frac{1}{N}\sum_i \log g^2(x_i) + \|g\|^2 + \frac{1}{N}\sum_i \log \lambda^2 && (26)\\
&= \lambda^{-1} \operatorname*{argmin}_{g\in\mathcal{H}} -\frac{1}{N}\sum_i \log g^2(x_i) + \|g\|^2 && (27)\\
&= \lambda^{-1} f. && (28)
\end{aligned}$$

■

Appendix G. Derivation of the Gradients, Proof Of Lemma 4

In this section we derive the expressions for standard and the natural gradients of the objective (5), as given in Lemma 4.

Proof [Proof Of Lemma 4] We first derive the expression for $\nabla_\alpha L$ in (18). Recall that $\|f\|_{\mathcal{H}}^2 = \langle \alpha, K\alpha\rangle_{\mathbb{R}^N}$ 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

$$\begin{aligned}
\nabla_\alpha \frac{1}{N}\sum_{i=1}^N \log f^2(x_i) &= \frac{1}{N}\sum_{i=1}^N f^{-2}(x_i)\cdot 2f(x_i)\cdot \nabla_\alpha f(x_i) && (29)\\
&= 2\frac{1}{N}\sum_{i=1}^N f^{-1}(x_i)\cdot \nabla_\alpha f(x_i) && (30)\\
&= 2\frac{1}{N} K (f(x_1), \ldots, f(x_N))^{-1} && (31)\\
&= 2\frac{1}{N} K \left(K\alpha\right)^{-1}. && (32)
\end{aligned}$$

This yields (18).

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

$$\nabla_f \frac{1}{N} \sum_{i=1}^{N} \log f^2(x_i) = \nabla_f \frac{1}{N} \sum_{i=1}^{N} \log \left\langle f, x_i \right\rangle_{\mathcal{H}}^2 \tag{33}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left\langle f, x_i \right\rangle_{\mathcal{H}}^{-2} \cdot 2 \left\langle f, x_i \right\rangle_{\mathcal{H}} \cdot \nabla_f \left\langle f, x_i \right\rangle_{\mathcal{H}} \tag{34}$$

$$= 2 \frac{1}{N} \sum_{i=1}^{N} \left\langle f, x_i \right\rangle_{\mathcal{H}}^{-1} x_i. \tag{35}$$

This completes the proof. ■

Appendix H. SDO Kernel Details

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

H.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

$$\left\| f \right\|_{\bar{a}}^2 = \sum_{l=0}^{m} a_l \sum_{|\kappa|_1 = l} \frac{l!}{\kappa!} \left\| (D^\kappa f) \right\|_{L_2}^2, \tag{36}$$

where D^κ are the κ-indexed derivative, as discussed in Section 4. 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,

$$\mathcal{H}^{\bar{a}} = \{ f \in L_2 \mid \left\| f \right\|_{\bar{a}} < \infty \} \tag{37}$$

and let the associated inner product be denoted by

$$\left\langle f, g \right\rangle_{\bar{a}} = \sum_{l=0}^{m} a_l \sum_{|\kappa|_1 = l} \frac{l!}{\kappa!} \left\langle (D^\kappa f), (D^\kappa g) \right\rangle_{L_2}. \tag{38}$$

Define the Fourier transform

$$\mathcal{F} f(z) = \int_{\mathbb{R}^d} f(u) e^{-2\pi i \langle z, u \rangle} du, \tag{39}$$

and recall that we have (see for instance [39], [11])

$$\mathcal{F}\left(D^\kappa f \right)(z) = \left(\prod_{j=1}^{d} (2\pi i z_j)^{\kappa_j} \right) \mathcal{F} f(z) \text{ for all } z \in \mathbb{R}^d. \tag{40}$$

The following connection between the L_2 and the derivative derived norms is well known for the standard Sobolev spaces ([28, 32, 44]). However, since (36) somewhat differs from the standard definitions, we provide the argument for completeness.

Lemma 6 *Set for $z \in \mathbb{R}^d$*

$$v_{\bar{a}}(z) = \left(1 + \sum_{l=1}^{m} a_l \cdot (2\pi)^{2l} \|z\|^{2l}\right)^{\frac{1}{2}}. \tag{41}$$

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

$$\|f\|_{\bar{a}}^2 = \|v_{\bar{a}}(z) \cdot \mathcal{F}[f]\|_{L_2}^2. \tag{42}$$

Proof

$$\begin{align}
\|f\|_{\bar{a}}^2 &= \sum_{l=0}^{m} a_l \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \|D^\kappa f\|_{L_2}^2 \tag{43}\\
&= \sum_{l=0}^{m} a_l \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \|\mathcal{F}\left[D^\kappa f\right]\|_{L_2}^2 \tag{44}\\
&= \int dz \left[\sum_{l=0}^{m} a_l \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \left|\mathcal{F}\left[D^\kappa f\right](z)\right|^2\right] \tag{45}\\
&= \int dz \left[\left|\mathcal{F}\left[f\right](z)\right|^2 + \sum_{l=1}^{m} a_l \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \left(\prod_{j=1}^{d} (2\pi z_j)^{2\kappa_j}\right) \left|\mathcal{F}\left[f\right](z)\right|^2\right] \tag{46}\\
&= \int dz \left|\mathcal{F}\left[f\right](z)\right|^2 \left[1 + \sum_{l=1}^{m} a_l \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \left(\prod_{j=1}^{d} (2\pi z_j)^{2\kappa_j}\right)\right] \tag{47}\\
&= \int dz \left|\mathcal{F}\left[f\right](z)\right|^2 \left[1 + \sum_{l=1}^{m} a_l \cdot (2\pi)^{2l} \sum_{|\kappa|_1=l} \frac{l!}{\kappa!} \prod_{j=1}^{d} z_j^{2\kappa_j}\right] \tag{48}\\
&= \int dz \left|\mathcal{F}\left[f\right](z)\right|^2 \left[1 + \sum_{l=1}^{m} a_l \cdot (2\pi)^{2l} \|z\|^{2l}\right] \tag{49}
\end{align}$$

■

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 6, for all $x \in \mathbb{R}^d$

$$\begin{align}
f(x) &= \left\langle f, k_x^{\bar{a}} \right\rangle_{\bar{a}} \tag{50}\\
&= \int_{\mathbb{R}^d} dz \;\; \mathcal{F}[f](z)\overline{\mathcal{F}[k_x^{\bar{a}}](z)} v_{\bar{a}}^2(z). \tag{51}
\end{align}$$

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

$$f(x) = \int dz \;\; \mathcal{F}[f](z) e^{2\pi i \langle x, z\rangle}. \tag{52}$$

This implies that

$$\int dz\ \ \mathcal{F}[f](z)e^{2\pi i\langle x,z\rangle} = \int_{\mathbb{R}^d} dz\ \ \mathcal{F}[f](z)\overline{\mathcal{F}[k_x^{\bar{a}}](z)}v_{\bar{a}}^2(z) \tag{53}$$

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

$$\mathcal{F}[k_x^{\bar{a}}](z) = \frac{e^{-2\pi i\langle x,z\rangle}}{v_{\bar{a}}^2(z)}. \tag{54}$$

Using Fourier inversion again we obtain

$$k^{\bar{a}}(x,y) = \int_{\mathbb{R}^d} \frac{e^{2\pi i\langle y-x,z\rangle}}{v_{\bar{a}}^2(z)}dz = \int_{\mathbb{R}^d} \frac{e^{2\pi i\langle y-x,z\rangle}}{1+\sum_{l=1}^m a_l\cdot(2\pi)^{2l}\left\|z\right\|^{2l}}dz. \tag{55}$$

H.2. Sampling Approximation

As discussed in Section 4.1, we are interested in sampling points $z \in \mathbb{R}^d$ from a finite non negative measure with density given by $w^a(z) = (1+a\cdot(2\pi)^{2m}\left\|z\right\|^{2m})^{-1}$. With a slight overload of notation, we will also denote by w_a the scalar function $w_a : \mathbb{R} \to \mathbb{R}$,

$$w^a(r) = (1 + a\cdot(2\pi)^{2m}r^{2m})^{-1}. \tag{56}$$

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 $f : \mathbb{R}^d \to \mathbb{C}$,

$$\int_{\mathbb{R}^d} w_a(z)f(z)dz = \int_0^\infty dr \int_{S^{d-1}} d\theta\ \ w_a(r)A_{d-1}(r)f(r\theta) \tag{57}$$

$$= A_{d-1}(1)\int_0^\infty dr \int_{S^{d-1}} d\theta\ \ \left[w_a(r)r^{d-1}\right]\cdot f(r\theta). \tag{58}$$

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

$$A_{d-1}(r) = \frac{2\pi^{d/2}}{\Gamma(d/2)}r^{d-1} \tag{59}$$

is the $d-1$ dimensional volume of the sphere or radius r in $\mathbb{R}^d$. The meaning of (58) is that to sample from w_a^{-2}, we can sample θ uniformly from the sphere (easy), and r from a density

$$\zeta(r) = w_a(r)r^{d-1} = \frac{r^{d-1}}{1+a\cdot(2\pi)^{2m}r^{2m}} \tag{60}$$

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 4.1, $\zeta(r)$ 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, [4]. 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.

Appendix I. A Few Basic Properties of the Kernel

Proposition 7 *The kernel* (10) *is real valued and satisfies*

$$K^a(x,y) = \int_{\mathbb{R}^d} \frac{\cos\left(2\pi\left\langle y-x, z\right\rangle\right)}{1 + a\cdot(2\pi)^{2m}\left\|z\right\|^{2m}} dz. \tag{61}$$

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

Proposition 8 *For all* $x, y \in \mathbb{R}^d$,

$$K^b(x,y) = b^{-\frac{d}{2m}} K^1(b^{-\frac{1}{2m}}x, b^{-\frac{1}{2m}}y) \tag{62}$$

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

$$\begin{aligned} K^b(x,y) &= \int_{\mathbb{R}^d} \frac{\cos\left(2\pi\left\langle y-x,z\right\rangle\right)}{1 + b\left\|2\pi\cdot z\right\|^{2m}} dz && (63)\\ &= \int_{\mathbb{R}^d} \frac{\cos\left(2\pi\left\langle b^{-\frac{1}{2m}}(y-x), u\right\rangle\right)}{1 + \left\|2\pi u\right\|^{2m}} du\cdot b^{-\frac{d}{2m}} && (64)\\ &= b^{-\frac{d}{2m}}K^1(b^{-\frac{1}{2m}}x, b^{-\frac{1}{2m}}y). && (65) \end{aligned}$$

■

Lemma 9 *There is a function* $c(m)$ *of* m *such that for every* $x \in \mathbb{R}^d$,

$$\int \left(K^a(x,y)\right)^2 dy = c(m)\cdot a^{-\frac{d}{2m}}. \tag{66}$$

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. (54)). We thus have

$$\begin{aligned} \|k_x^a\|_{L_2}^2 &= \|\mathcal{F}(k_x^a)\|_{L_2}^2 && (67)\\ &= \int \frac{e^{-2\pi i\langle x,z\rangle}\cdot\overline{e^{-2\pi i\langle x,z\rangle}}}{\left(1 + a(2\pi)^{2m}\left\|z\right\|^{2m}\right)^2} dz && (68)\\ &= \int \frac{1}{\left(1 + a(2\pi)^{2m}\left\|z\right\|^{2m}\right)^2} dz && (69)\\ &= a^{-\frac{d}{2m}} \int \frac{1}{\left(1 + (2\pi)^{2m}\left\|z'\right\|^{2m}\right)^2} dz', && (70)\\ & && (71) \end{aligned}$$

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

Appendix J. KDE vs RSR Comparison Proofs

In this section we develop the ingredients required to prove Proposition 3. In section J.1 we reduce the solution of the RSR 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 J.2 we use these results to prove Proposition 3.

J.1. Solution Of RSR for a 2-Block Model

As discussed in section B, any RSR solution f must be a zero point of the natural gradient, $\nabla_f L = 0$. Using the expressions given following Lemma 4, 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 $\beta = (a, \ldots, a, b, \ldots, b)$, where a is in first N coordinates and b is in the next M. Then, as mentioned in section B, $\beta = (K\beta)^{-1}$ 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 RSR 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,

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

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

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

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

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

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

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

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

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

■

J.2. Proof Of Proposition 3

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 RSR 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 $\rho \in \{+1, -1\}$. ■

Appendix K. 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 2.1, 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 (18) for the natural gradient is given in Lemma 4.

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. ■

Appendix L. 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

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

where $p_m(x;\tau)$ is a possibly unnormalized probability density on $\mathbb{R}^d$, evaluated at $x \in \mathbb{R}^d$, and dependent on the hyperparameter τ. 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.

In this work, we employ this concept, leveraging it to identify the choice of parameters (in our case, τ, 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* ([17]), a particular approach to measuring the Fisher divergence between a dataset sampled from an unknown distribution and a proposed distribution model. The full details of the procedure are in Supllmentry Metrial Section N

L.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, $\tilde{p_m}(x;\tau)$ (where τ is a parameter). Score matching sets out to reduce the Fisher divergence between p_d and $\tilde{p_m}(\cdot;\tau)$, formally expressed as

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

As detailed in [17], the technique of integration by parts can derive an expression that does not depend on the unknown latent score function s_d:

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

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

L.2. The Hessian Estimation for Small τ's

Deriving the Hessian for small τ values proves to be challenging. Note that small τ 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 τ. In this context, we define a stable local minimum as a point preceded and succeeded by three points, each greater than the focal point.

L.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 [16], 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 [12] 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 $\nabla_x s(x_i) = \frac{1}{h} \cdot \left(s\left(x_i + h \cdot \epsilon\right) - s\left(x_i\right)\right)$ 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 $n$
Initialize $traceEstimator$ to 0
for $i = 1$ to $n$ do
    Sample random vector $\epsilon$ from normal distribution
    Calculate $s(\tau; x + h * \epsilon)$
    Calculate $(s(\tau; x + h * \epsilon) - s(\tau; x))$
    Compute $(1/h) \cdot (s(\tau; x + h * \epsilon) - s(\tau; x)) \cdot \epsilon$
    Add result to $traceEstimator$
end for
Return $\frac{traceEstimator}{n}$
```

Appendix M. Experiments

M.1. AUC-ROC Performance Analysis

M.2. Anomaly Detection Results for ADbench

This section presents an evaluation of our approach on real-world tasks and data, focusing on Anomaly Detection (AD) where normalized density is not a concern. AD was chosen for evaluation due to its inherent attributes that align closely with density estimation, including the differentiation of samples from the latent and out-of-distribution. We compare our results to a golden standard AD benchmark, ADbench ([14]), that evaluates a wide range of 15 AD algorithms on over 47 labeled datasets. In addition, we evaluate KDE using both Gaussian and Laplace kernels, and as an ablation study, we compare RSR to KDE with SDO kernel.

We focus on the unsupervised setup, in which no labeled anomalies are given in the training phase. For all density-based approaches, we employ the negative of the density as the 'anomaly score'. The ADbench paper evaluates success on each dataset using AUC-ROC. In addition to AUC-ROC, we also focus on a ranking system as follows: for each dataset, we convert raw AUC-ROC scores of the methods into rankings from 1 to 18. Here, 18 denotes the best performance on a given dataset, and 1 the worst. This mitigates bias inherent in averaging AUC-ROC scores themselves across datasets,

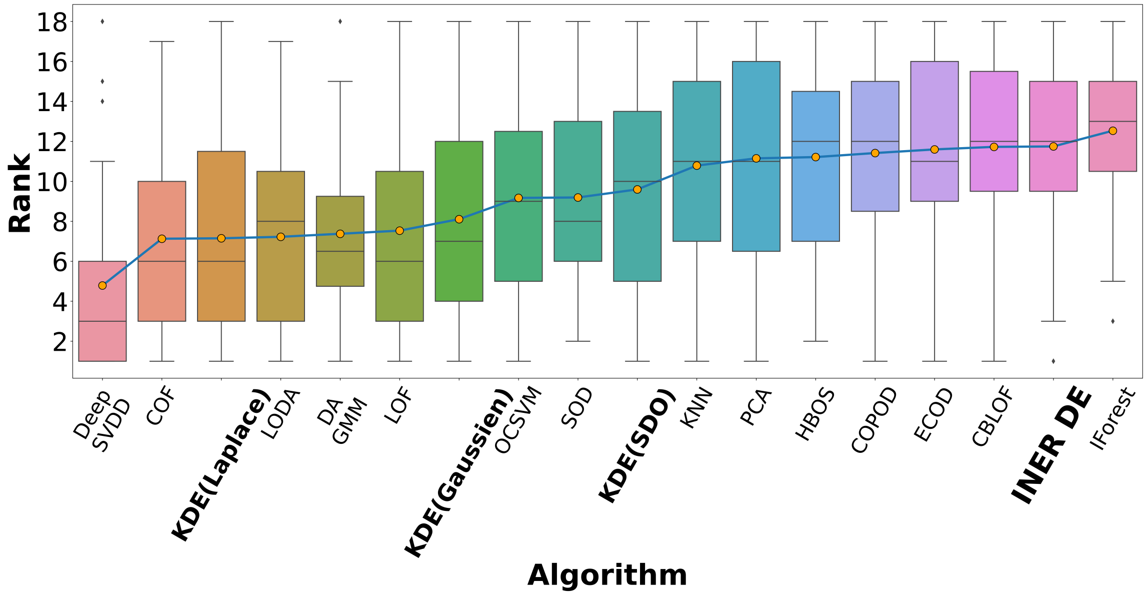

Figure 4: Anomaly Detection Results on ADBench. Relative Ranking Per Dataset, Higher is Better. RSR is Second Best Among 18 Algorithms

due to generally higher AUC-ROC scores on easier datasets. This is important since no single AD method consistently outperforms others in all situations, as discussed in detail in [14].

For both AUC-ROC and rank evaluations, **RSR emerges as the 2nd best AD method overall**. Notably, this achievement is with the 'vanilla' version of our method, without any pre or post-processing dedicated to AD. In contrast, many other methods are specifically tailored for AD and include extensive pre and post-processing. In Figure 4, for each algorithm we present the box plot with the average ranking over all datasets (along with quantiles). The algorithms are sorted by the average ranking. A similar plot for raw AUC-ROC values is given in the supplementary material, and it presents a similar picture. 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.

In addition to performing well on the standard ADBench benchmark, and perhaps even more impressively, RSR excels also on the more demanding setup of *duplicate anomalies*, which was also extensively discussed in [14]. Here, **RSR 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 such as equipment failures, is a focal point for ADbench's assessment of unsupervised anomaly detection methods due to its inherent difficulty, leading to substantial drops in performance for former leaders like Isolation Forest.

In Section 5 we presented a comparative study of various methods according to their ranking across different datasets. This section provides an analysis of the raw AUC-ROC values themselves.

We present two figures to elucidate our findings. Figure 5 is similar to the box plot from shown in Figure 4 in the main text but includes raw AUCROC values instead of rankings. As the figure shows, RSR also secures the second-highest position in terms of average raw AUC-ROC (second only to IForest), as represented by the order of methods on the X-axis. Furthermore, the median line within the boxes indicates that RSR is the method with the highest median score. Figure 6 shows a heatmap representation of the AUC-ROC values. In this visualization, the size of the circle symbolizes the corresponding AUC value, while the color gradient signifies the deviation in AUC value from RSR. The purpose of this heatmap is to offer a graphical interpretation of the AUC-ROC performance levels, demonstrating how they diverge from the performance of RSR.

M.3. Duplicate Anomalies.

Duplicate anomalies are often encountered in in various applications due to factors such as recording errors [22], a circumstance termed as "anomaly masking" [5, 13], posing significant hurdles for diverse AD algorithms. The significance of this factor is underscored in ADbench [14], 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 RSR 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 RSR possessing the highest median and its robustness to duplicate anomalies, along with a probable resistance to a high number of anomalies, not only emphasizes its superiority as a DE method but also underscores its potential to serve as a state-of-the-art AD method.

M.4. Hyperparameter Tuning with On-the-fly Fisher-Divergence Minimization

A primary task at hand involves hyperparameter tuning to select the optimal τ according to the Fisher-Divergence (FD) procedure, as detailed in section L and Section 5. For efficient computation, we employ an on-the-fly calculation approach. The process initiates with a τ value that corresponds to the 'order'-th largest tested point. Subsequently, we explore one larger and one smaller τ 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.

Appendix N. 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 (11)) $2.n_{iters}$, the number of iterations used in the gradient decent approximating the optimization for α. $3.lr$, the learning rate applied in the gradient decent α 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.

Algorithm 2: Density Estimation Procedure with Hypeparameter Tuning

```
repeat
    Estimate the SDO kernel matrix via sampling as described in Section 2 and detailed in Algorithm 3.
    Determine the optimal $\alpha$ as described in Section 2 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 2 and detailed in Algorithm 5.
    Assess the Fisher-divergence as described in Appendix Section L and detailed in both Algorithm 6 and Algorithm 1.
until For each smoothness parameter $\tau$.
The density estimator $F^2$ corresponds to the $\tau$ value that yields the minimal FD.
```

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

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

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

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

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

```
Require: $X, Y$
$\alpha \leftarrow Optimal\_alpha(X, \ldots)$
$e \leftarrow K(X, Y)$
return $(e \cdot \alpha)^2$
```

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 ← $\nabla \sum \log(F^2(X,Y))$
traces_sum ← $\mathbf{0}$
for $i = 1$ to $n_{\text{iters}}$ do
    $\epsilon \sim \mathcal{U}(\{-1,0,1\}^{\text{size}(X)})$
    shifted_scores ← $\nabla \sum \log(F^2(X, Y + h \cdot \epsilon))$
    traces_sum ← traces_sum + $\sum((\text{shifted\_scores} - scores) \cdot \epsilon)/h$
end for
traces ← traces_sum/$n_{\text{iters}}$
return $\mathbb{E}[\text{traces} + \frac{1}{2} \cdot \|\textbf{scores}\|^2]$
```

Appendix O. Consistency Theorem

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

Recall that for any $a > 0$, $\mathcal{H}^a$ is the RKHS with the norm given by (8) and the kernel $k^a(x,y)$ given by (10).

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

$$L(f) = L^a(f) = \frac{1}{2}\left(-\frac{1}{N}\sum \log f^2(x_i) + \|f\|^2_{\mathcal{H}^a}\right). \tag{105}$$

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 D and that L is convex when restricted to the open cone

$$\mathcal{C}' = (\mathcal{C}^a)' = \left\{ f \in span\,\{k_{x_i}\}_1^N \;\mid\; f(x_i) > 0 \right\}. \tag{106}$$

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 D, 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 (105) in the cone (106). 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 RSR 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 O.1 below we provide an overview of the proof, and in Section O.2 full details are provided.

O.1. Overview Of The Proof

As discussed in Section A, consistency for the 1 dimensional case was shown in [21] 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 (105), given by Lemma 13 below. In this Lemma we observe that L is strongly convex and thus for any function v, we have

$$\|u - v\|_{\mathcal{H}^a} \leq \|\nabla L(v)\|_{\mathcal{H}^a}, \tag{107}$$

where u is the true minimizer of L in $\mathcal{C}'$ (i.e. the RSR 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 D, the gradient of L in $\mathcal{H}^a$ is given by

$$\nabla L(f) = -\frac{1}{N}\sum f(x_i)^{-1}k_{x_i} + f. \tag{108}$$

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

$$\|\nabla L(v)\|_a^2 = \frac{1}{N^2}\sum_{i,j} v^{-1}(x_i)v^{-1}(x_j)k(x_i,x_j) - 2\frac{1}{N}\sum_i v^{-1}(x_i)\left\langle k_{x_i}, v\right\rangle_a + \|v\|_a^2 \tag{109}$$

$$= \frac{1}{N^2}\sum_{i,j} v^{-1}(x_i)v^{-1}(x_j)k(x_i,x_j) - 2 + \|v\|_a^2. \tag{110}$$

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 (110) concentrates around 1. To this end, we will first show that the expectation of

$$\frac{1}{N^2}\sum_{i,j} v^{-1}(x_i)v^{-1}(x_j)k(x_i,x_j) \tag{111}$$

(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 RSR 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 (111) around its mean. This requires the control of cross products of the terms in (111) 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.

O.2. Full Proof Details

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

Next, we observe that $L(f)$ is *strongly* convex with respect to the norm $\|\cdot\|_{\mathcal{H}^a}$ (see [27] 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 $v \in \mathcal{C}'$,*

$$\|u - v\|_{\mathcal{H}^a} \leq \|\nabla L(v)\|_{\mathcal{H}^a} . \tag{112}$$

Proof The function $f \mapsto \|f\|^2_{\mathcal{H}^a}$ is 2-strongly convex with respect to $\|\cdot\|^2_{\mathcal{H}^a}$. 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

$$\langle \nabla L(v) - \nabla L(u), v - u \rangle_a \geq \|v - u\|^2_a , \tag{113}$$

see [27], 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

$$\|\nabla L(v)\|_a \cdot \|u - v\|_a \geq \langle \nabla L(v), v - u \rangle_a \geq \|v - u\|^2_a , \tag{114}$$

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:

$$\|\nabla L(v)\|^2_a = \frac{1}{N^2} \sum_{i,j} v^{-1}(x_i) v^{-1}(x_j) k(x_i, x_j) - 2 \frac{1}{N} \sum_i v^{-1}(x_i) \langle k_{x_i}, v \rangle_a + \|v\|^2_a \tag{115}$$

$$= \frac{1}{N^2} \sum_{i,j} v^{-1}(x_i) v^{-1}(x_j) k(x_i, x_j) - 2 + \|v\|^2_a . \tag{116}$$

Since v is a fixed function in $\mathcal{H}^1$, it is clear from definition (8) that $\|v\|^2_a - 1 \to 0$ as $a \to 0$. Thus, to bound $\|\nabla L(v)\|^2_a$, it suffices to bound

$$\frac{1}{N^2} \sum_{i,j} v^{-1}(x_i) v^{-1}(x_j) k(x_i, x_j) - 1 \tag{117}$$

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

Note that the form (10) of the kernel k^a implies that this is a stationary kernel, i.e. $k^a(x, y) = g^a(x - y)$ with

$$g^a(x) = \int_{\mathbb{R}^d} \frac{e^{2\pi i \langle -x, z \rangle}}{1 + a \cdot (2\pi)^{2m} \|z\|^{2m}} dz. \tag{118}$$

Lemma 14 *Let $k^a(x, y)$ be the SDO kernel defined by (10). For any function $v \in L_2$, and $x \in \mathbb{R}^d$ set*

$$(K_a v)(x) = \int k^a(x, y) v(y) dy. \tag{119}$$

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 (119), 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 H.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

$$\|K_a v - v\|_{L_2}^2 = \|\mathcal{F}(K_a v - v)\|_{L_2}^2 \tag{120}$$

$$= \left\| \mathcal{F}(v) \cdot \left(\frac{1}{1 + a\cdot(2\pi)^{2m}\|z\|^{2m}} - 1 \right) \right\|_{L_2}^2 \tag{121}$$

$$= \left\| \mathcal{F}(v) \cdot \left(\frac{a\cdot(2\pi)^{2m}\|z\|^{2m}}{1 + a\cdot(2\pi)^{2m}\|z\|^{2m}} \right) \right\|_{L_2}^2 \tag{122}$$

$$= \int |\mathcal{F}(v)(z)|^2 \cdot \left(\frac{a\cdot(2\pi)^{2m}\|z\|^{2m}}{1 + a\cdot(2\pi)^{2m}\|z\|^{2m}} \right)^2 dz \tag{123}$$

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 (123) 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

$$\int |\mathcal{F}(v)(z)|^2 \cdot \left(\frac{a\cdot(2\pi)^{2m}\|z\|^{2m}}{1 + a\cdot(2\pi)^{2m}\|z\|^{2m}} \right)^2 dz \tag{124}$$

$$= \int_{B(r)} |\mathcal{F}(v)(z)|^2 \cdot \left(\frac{a\cdot(2\pi)^{2m}\|z\|^{2m}}{1 + a\cdot(2\pi)^{2m}\|z\|^{2m}} \right)^2 dz + \int_{B^c(r)} |\mathcal{F}(v)(z)|^2 \cdot \left(\frac{a\cdot(2\pi)^{2m}\|z\|^{2m}}{1 + a\cdot(2\pi)^{2m}\|z\|^{2m}} \right)^2 dz \tag{125}$$

$$\leq \varepsilon \cdot \|v\|_{L_2}^{-1} \int_{B(r)} |\mathcal{F}(v)(z)|^2 dz + \int_{B^c(r)} |\mathcal{F}(v)(z)|^2 dz \tag{126}$$

$$\leq \varepsilon + \varepsilon. \tag{127}$$

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 (117) 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 (118). 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 [23], 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*

$$\frac{g^{a(N)}(0)}{N^2}\sum_i v^{-2}(x_i) \to 0 \tag{128}$$

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

$$\mathbb{E}v^{-2}(x) = \int v^{-2}(x)\cdot v^2(x)\mathbb{1}_{\{B\}}(x)dx = \lambda(B) < \infty. \tag{129}$$

Thus the Marcinkiewicz-Kolmogorov law implies that

$$\frac{1}{N}\sum_i v^{-2}(x_i) \to \lambda(B) \tag{130}$$

almost surely. Next, recall that by Proposition 8, 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 (130) 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 (119). We have

$$\begin{aligned}
|\mathbb{E}Y_{ij} - 1| &= \left|\int v^{-1}(x)v^{-1}(y)k^a(x,y)v^2(x)v^2(y)dxdy - \int v^2(x)dx\right| && (131)\\
&= \left|\int k^a(x,y)v(x)v(y)dxdy - \int v^2(x)dx\right| && (132)\\
&= \left|\int dx\cdot v(x)\left[(K_a v)(x) - v(x)\right]\right| && (133)\\
&\leq \|v\|_{L_2}\|(K_a v) - v\|_{L_2} && (134)\\
&= \|(K_a v) - v\|_{L_2}. && (135)
\end{aligned}$$

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

$$|\mathbb{E}Y_{ij}| = \langle K_a v, v\rangle_{L_2} \leq \|K_a v\|_{L_2}\|v\|_{L_2} \leq 1, \tag{136}$$

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,

$$\frac{1}{N^4}\mathbb{E}\left(\sum_{i\neq j}(Y_{ij} - EY_{ij})\right)^2 = \frac{1}{N^4}\sum_{i\neq j, i'\neq j'}\left(\mathbb{E}Y_{ij}Y_{i'j'} - \mathbb{E}Y_{ij}\mathbb{E}Y_{i'j'}\right) \tag{137}$$

and then using the Chebyshev inequality. Observe that there are three types of terms of the form $\left(\mathbb{E}Y_{ij}Y_{i'j'} - \mathbb{E}Y_{ij}\mathbb{E}Y_{i'j'}\right)$ on the right hand side of (137). 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*

$$\mathbb{E}Y_{ij}^2 \leq \lambda(B) a^{-\frac{d}{2m}} c(m) < \infty. \tag{138}$$

Proof Write

$$\begin{aligned}
\mathbb{E}Y_{ij}^2 &= \int v^{-2}(x) v^{-2}(y) (k^a(x,y))^2 v^2(x) v^2(y) dx dy &(139)\\
&= \int \mathbb{1}_{\{B\}}(x) \mathbb{1}_{\{B\}}(y) (k^a(x,y))^2 dx dy &(140)\\
&\leq \int \mathbb{1}_{\{B\}}(x) \left\| k_x^a \right\|_{L_2}^2 dx &(141)\\
&= \lambda(B) a^{-\frac{d}{2m}} c(m), &(142)
\end{aligned}$$

where we have used Lemma 9 to compute $\|k_x^a\|_{L_2}^2$. ■

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

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

$$|\mathbb{E}Y_{ij}Y_{jt}| \leq 1. \tag{143}$$

Proof

$$\mathbb{E}Y_{ij}Y_{jt} = \int v^{-1}(x_i) v^{-1}(x_j) k^a(x_i,x_j) v^{-1}(x_j) v^{-1}(x_t) k^a(x_j,x_t) v^2(x_i) v^2(x_j) v^2(x_t) dx_i dx_j dx_t \tag{144}$$

$$\begin{aligned}
&= \int v(x_i) k^a(x_i,x_j) v(x_t) k^a(x_t,x_j) \mathbb{1}_{\{B\}}(x_j) dx_i dx_j dx_t &(145)\\
&= \int (K_a v)^2(x_j) \mathbb{1}_{\{B\}}(x_j) dx_j &(146)\\
&\leq \int (K_a v)^2(x_j) dx_j &(147)\\
&= \|K_a v\|_{L_2}^2 &(148)\\
&\leq 1, &(149)
\end{aligned}$$

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*

$$\mathbb{E}(Y_{ij} - EY_{ij})(Y_{i'j'} - EY_{i'j'}) = 0. \tag{150}$$

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

We now collect these results to bound (137).

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

$$\frac{1}{N^4}\mathbb{E}\left(\sum_{i\neq j}(Y_{ij} - EY_{ij})\right)^2 \le \frac{c'(m, B)}{N} \tag{151}$$

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 $a^{-\frac{d}{2m}} = a^{-\frac{d}{2m}}(N) \le N$, and thus, the overall contribution of such terms to the sum in (151) 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.

Proof [Of Theorem 12] First, observe that by definition $\|u - v\|_{L_2} \le \|u - v\|_a$ for any $u, v \in \mathcal{H}^1$. Next, by Lemma 13 and using (116), we have

$$\|u_N - v\|_{L_2} \le \|u_N - v\|_a \le \frac{1}{N^2}\sum_{i,j} v^{-1}(x_i)v^{-1}(x_j)k(x_i, x_j) - 2 + \|v\|_a^2 . \tag{152}$$

Clearly, by definition (8), we have $\|v\|_a^2 - 1 \to 0$ as $a \to 0$. Write

$$\frac{1}{N^2}\sum_{i,j} v^{-1}(x_i)v^{-1}(x_j)k(x_i, x_j) - 1 = \frac{1}{N^2}\sum_{i} Y_i + \frac{1}{N^2}\sum_{i\neq j}(Y_{ij} - \mathbb{E}Y_{ij}) + \left[\frac{N^2 - N}{N^2} \cdot \mathbb{E}Y_{12} - 1\right] . \tag{153}$$

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. ■

Appendix P. 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
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

0.45
(*a*) b

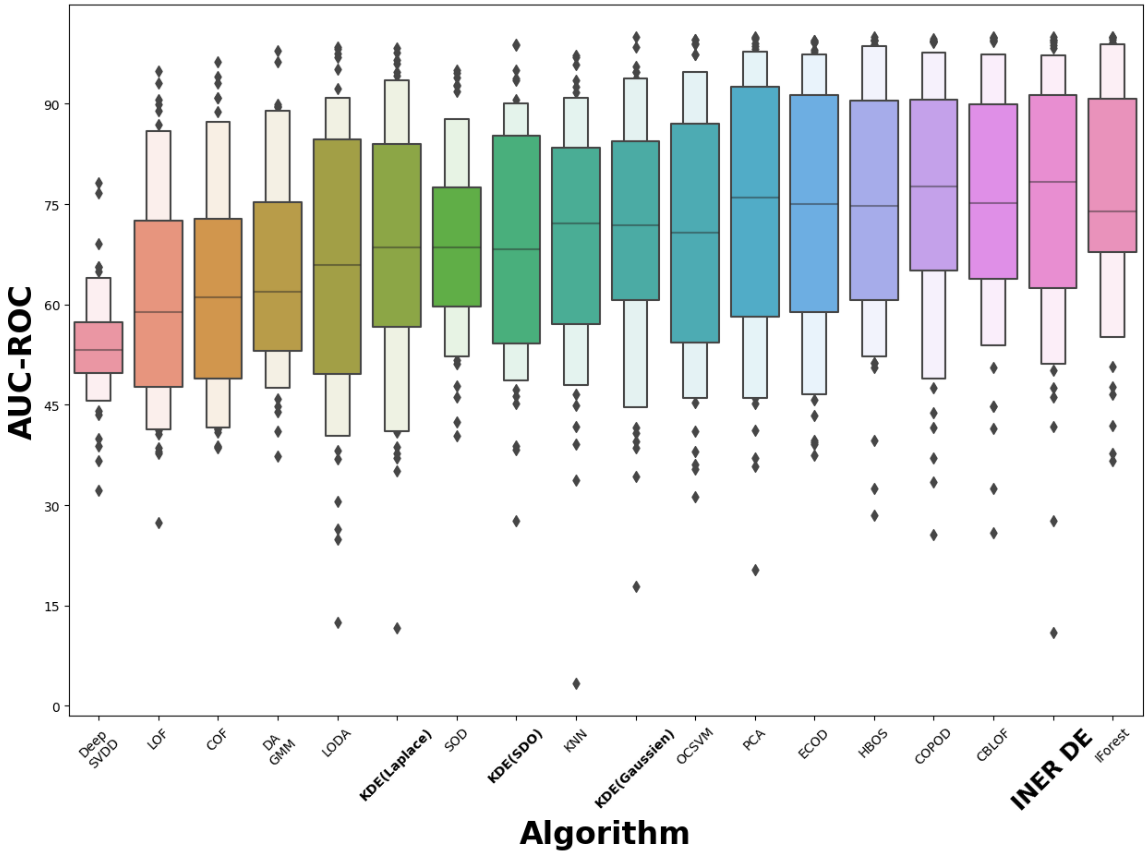

Figure 5: Box plot presenting actual AUC-ROC values.

0.45
(*a*) b

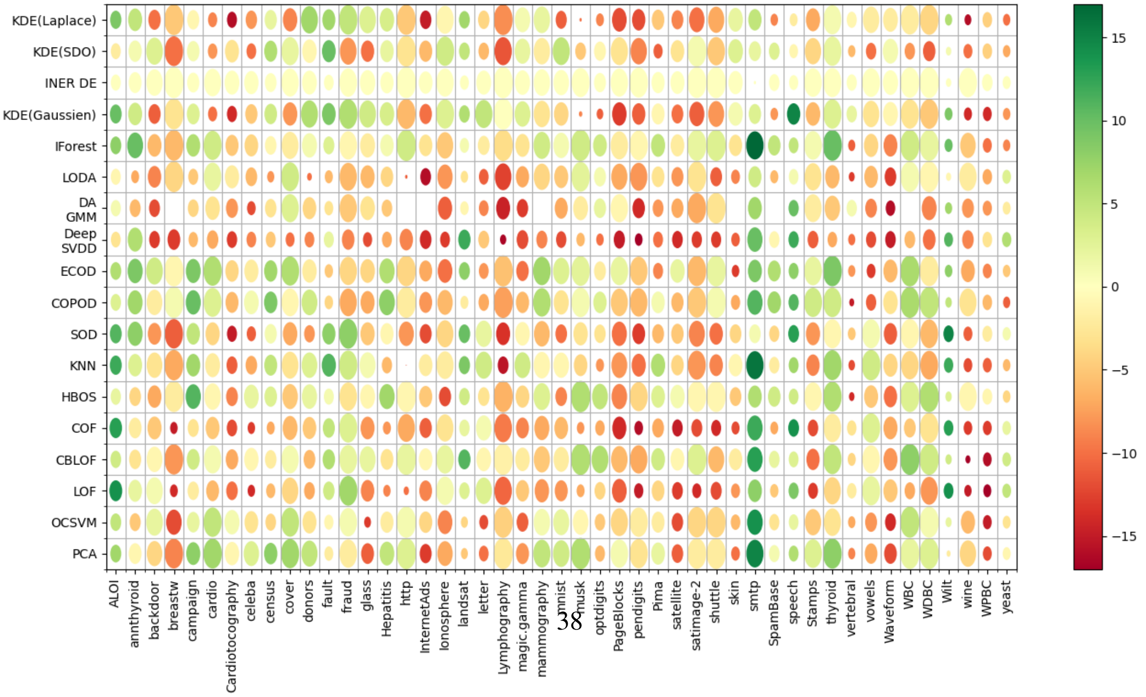

Figure 6: Heatmap of AUC-ROC values. Circles size represent absolute value, color id the shift from RSR.