Locality Sensitive Hashing for p-Exponential Kernels with Applications to Density Estimation

Barak Gorodissky
Tel Aviv University
gorodissky@mail.tau.ac.il

Tal Wagner
Tel Aviv University
talwag@tauex.tau.ac.il

Abstract

A kernel $\mathbf{k}(x, y)$ is *LSHable* if there exists a locality sensitive hashing scheme $\mathcal{H}$ such that $\mathbf{k}(x, y) = \Pr_{h\sim\mathcal{H}}[h(x) = h(y)]$ for all x, y. This notion plays a key role in efficient kernel methods in high dimensions. In this work, we show that the p-exponential kernel $\mathbf{k}(x, y) = \exp(-\|x - y\|_p)$ is LSHable in bounded regions for all $1 < p \leq 2$. Previously, this was known only for $p = 1$. Our new *mosaic LSH* scheme is based on a Poisson hyperplane process with hyperplanes sampled as ℓ_1-biased p-stable vectors, for which we develop efficient sampling procedures. As applications, our results yield new and efficient density estimation methods based on LSHability for those p-exponential kernels.

1 Introduction

Kernel densities have long been fundamental to machine learning [21]. Hashing-based techniques have recently emerged as a powerful approach for kernel in high-dimensional settings where traditional techniques are often rendered infeasible. The central notion is the following:

Definition 1.1 (kernel LSH)**.** *Let* $\mathbf{k} : \mathcal{X} \times \mathcal{X} \to [0, 1]$ *be a map over pairs in* $\mathcal{X}$*. A* locality sensitive hashing (LSH) scheme *for* $\mathbf{k}$ *is a distribution* $\mathcal{H}$ *over hash functions on* $\mathcal{X}$ *such that*

$$\forall\, x, y \in \mathcal{X}, \qquad \mathbf{k}(x, y) = \Pr_{h\sim H}[h(x) = h(y)]. \tag{1}$$

$\mathcal{H}$ must be endowed with an algorithm for sampling $h \sim \mathcal{H}$, a finite representation of h, and an algorithm to evaluate $x \mapsto h(x)$ given $x \in \mathcal{X}$ and h. If such $\mathcal{H}$ exists, we call $\mathbf{k}$ *an* LSHable kernel.

This notion was studied in Charikar's seminal work on the SimHash algorithm [14], and more recently has been central in high-dimensional kernel density estimation (KDE) due to a breakthrough work by Charikar and Siminelakis [13]. Intuitively, $\mathbf{k}$ encodes the similarity between pairs of elements in $\mathcal{X}$, and $\mathcal{H}$ is "locality sensitive" because similar elements are more likely to be hashed together. This property is the driving force behind a range of efficient high-dimensional kernel methods, including relative-error KDE [13, 44, 6], streaming KDE [16, 32], private KDE [17, 48, 47], kernel ridge regression [50], partition function estimation for importance sampling [45], and long-context attention in large language models [26, 49].

Despite its centrality and usefulness, there are rather few kernels for which LSHability is known. Classical examples include MinHash [10] for the Jaccard similarity between sets and SimHash [14] for the angular similarity between points on a sphere. For kernels over $\mathbb{R}^d$, the Random Binning Features scheme of Rahimi and Recht [39] constitutes an LSH scheme for the 1-exponential kernel, a.k.a the Laplacian kernel, $\mathbf{k}(x, y) = \exp(-\|x - y\|_1)$. Backurs et al. [6] gave a different LSH scheme for the same kernel in a bounded region. This has left as an open problem even the natural case of the 2-exponential kernel,[1] $\mathbf{k}(x, y) = \exp(-\|x-y\|_2)$. Applications of LSH-based methods to the 2-exponential and to other kernels have largely gone through various approximations or heuristics.

[1] Note that the 2-exponential kernel is different from the Gaussian kernel, $\mathbf{k}(x, y) = \exp(-\|x - y\|_2^2)$, in which the ℓ_2 norm in the exponent is squared. We show in Appendix C that no LSH scheme exists for the Gaussian kernel.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Summary of LSH results for ℓ_p metrics and p-exponential kernels.

	$p = 1$	$p \in (1, 2)$	$p = 2$
Metric LSH for $\|x - y\|_p$	[23]	[18, 36]	[18, 3]
Kernel LSH for $\exp(-\|x - y\|_p)$	[39, 6]	This work	This work

Our results. In this work, we design new kernel LSH schemes and prove the LSHability property for all p-exponential kernels with $1 < p \leq 2$ in a bounded region. Our main result is the following.

Theorem 1.2. *Let $p \in (1, 2]$ and $\sigma > 0$. There exists a kernel LSH scheme $\mathcal{H}$ that satisfies Definition 1.1 for the p-exponential kernel, $\mathbf{k}(x, y) = \exp(-\|x - y\|_p/\sigma)$, over points in $\mathbb{R}^d$ with coordinates bounded in $[-1, 1]$. The expected time to sample a hash function $h \sim \mathcal{H}$ and to evaluate h on a point x are each $O(d^2/\sigma)$.*

The coordinate restriction to $[-1, 1]$ can be scaled to any interval by inversely scaling the kernel bandwidth σ. Thus, Theorem 1.2 provides a kernel LSH scheme for any bounded region. The running time of the LSH scheme grows linearly with the side-length of a bounding hypercube for the region.

As applications, Theorem 1.2 leads to new LSH-based algorithms for p-exponential kernels. We highlight two examples in Section 3: relative-error KDE and differentially private KDE.

1.1 Background and Related Work

Metric LSH. LSH was originally introduced by Indyk and Motwani [23] for metric spaces in the following form, for the purpose of approximate nearest neighbor search in high dimensions.

Definition 1.3 (metric LSH [23])**.** *Let $(\mathcal{X}, \mathrm{D})$ be a metric space. A (r_1, r_2, p_1, p_2)-*LSH *scheme for $(\mathcal{X}, \mathrm{D})$ is a distribution $\mathcal{H}$ over hash functions on $\mathcal{X}$ such that for each pair $x, y \in \mathcal{X}$,*

$$\mathrm{D}(x, y) \leq r_1 \Rightarrow \Pr_{h \sim \mathcal{H}}[h(x) = h(y)] \geq p_1 \quad \textit{and} \quad \mathrm{D}(x, y) > r_2 \Rightarrow \Pr_{h \sim \mathcal{H}}[h(x) = h(y)] \leq p_2.$$

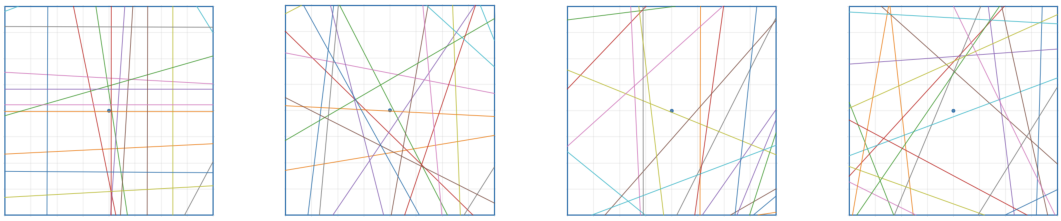
They designed metric LSH schemes for ℓ_1 and ℓ_2 metric spaces. Datar et al. [18] designed metric LSH schemes for all ℓ_p metrics with $p \in [1, 2]$. Their approach relies on p-stable distributions, which generalize the spherical invariance (i.e., 2-stable) property of the Gaussian distribution [22] (see Section 2.1). Andoni and Indyk [3] later presented a dramatically improved metric LSH for ℓ_2, which turned out to be optimal [38, 35]. Their approach was extended to $p \in (1, 2)$ by Nguyen [36].

The interest in ℓ_p metrics with $p \in (1, 2)$ has both mathematical and empirical motivations. Mathematically, these spaces embed isometrically into an infinite-dimensional L_1 space, suggesting that algorithmic results known for ℓ_1 might extend to them. In machine learning, ℓ_p-based formulations offer a useful interpolation between sparsity-promoting ℓ_1 formulations and smooth ℓ_2 formulations, with empirical benefits demonstrated in several applications [28, 41, 31, 46, 29]..

Kernel LSH. Charikar [14] focused on kernel LSH as defined in Definition 1.1, which had been previously used for set similarity measures like Broder's MinHash algorithm for the Jaccard similarity [10]. He showed that kernel LSH is achievable and useful for vectors in $\mathbb{R}^d$, designing the seminal SimHash algorithm. He also showed that if $\mathbf{k}(x, y)$ is LSHable then $1 - \mathbf{k}(x, y)$ must satisfy the triangle inequality. Further properties of kernel LSH were studied by Chierichetti and Kumar [15].

Rahimi and Recht [39] designed the Random Binning Features method for the 1-exponential (a.k.a. Laplacian) kernel. As observed by Andoni and Indyk in [2], this method constitutes a kernel LSH scheme. Backurs et al. [6] gave a different kernel LSH scheme for the same kernel. Their scheme is restricted to points in a bounded region and is faster in certain parameter regimes.

Both of these LSH schemes for the 1-exponential kernel share the same underlying technique: imposing a random axis-aligned grid on $\mathbb{R}^d$ and hashing each point to the grid cell that contains it. They differ in how the random grid is selected. The randomized grid approach is well-suited for ℓ_1, albeit less so for other ℓ_p spaces. The metric LSH schemes for $p \in [1, 2]$ by [18] are also based on randomly shifted grids, composed over projections onto p-stable directions. Those projections can be viewed as lossy probabilistic embeddings of ℓ_p into ℓ_1. This approach suffices for metric LSH, which implicitly involves an approximation gap (the gap is $c = r_2/r_1$ in Definition 1.3). Unfortunately, it does not yield the exact collision identity in Definition 1.1 outside the $p = 1$ case. Therefore, new



$p = 1.25$ $p = 1.5$ $p = 1.75$ $p = 2$

Figure 1: Samples from the mosaic LSH in two dimensions for various values of p.

techniques are needed to achieve kernel LSH for $p \in (1, 2]$. Our mosaic LSH scheme departs from randomly shifted grids and relies on Poisson hyperplane processes.. Table 1 contains a summary of metric and kernel LSH results for $p \in [1, 2]$.

Approximate kernel LSH. Approximate variants of kernel LSH relax the equality in (1) to hold only up to an additive [2, 48] or multiplicative [13, 6] approximation factor. Such variants suffice for some LSHability-based methods, with an additional loss depending on the approximation quality.

A prominent example is the 2-exponential kernel $\exp(-\|x - y\|_2)$. An approximate kernel LSH scheme can be obtained by composing a $\ell_2 \to \ell_1$ embedding step with distortion $1 \pm \varepsilon$, based on a variant of the Johnson–Lindenstrauss (JL) transform [1, 33], with a kernel LSH scheme for $\exp(-\|x - y\|_1)$. This approach has been used in [13, 6, 48] to apply LSHability methods to the 2-exponential kernel. The embedding distortion parameter ε controls the discrepancy between the collision probability and the target kernel, and the sampling and evaluation costs depend polynomially on ε^{-1}. It therefore offers a trade-off between approximation accuracy and computational cost. For $p \in (1, 2)$, approximate $\ell_p \to \ell_1$ embeddings (e.g., [25]) suggest an analogous route to approximate kernel LSH for p-exponential kernels, if they can be made algorithmic. However, such schemes would retain the embedding distortion as a parameter. Our results instead establish exact kernel LSH for all $p \in (1, 2]$, removing the embedding distortion as a source of error and computational cost.

Random tessellations. Balog et al. [8] showed that the pointwise cell collision probability of the Mondrian partition process [40] is equal to the Laplacian kernel. O'Reilly and Tran [37] extended this to the 2-exponential kernel via an isotropic STIT process. These results establish stochastic processes with the collision identity in eq. (1), but without the sampling and evaluation bounds needed for LSH.

1.2 Our Techniques

The first main step in our new approach to kernel LSH is to simulate Poisson hyperplane processes (PHPs). These are stochastic processes for generating random hyperplanes in $\mathbb{R}^d$. The sampled hyperplanes induce a partition of $\mathbb{R}^d$ called a *mosaic* [42]. The mosaic cells will be our LSH bins.

The appeal of PHPs in LSH for p-exponential kernels is the following. For a compact subset $A \subset \mathbb{R}^d$ let $\mathbb{H}(A)$ denote the set of all affine hyperplanes that intersect it. The number of samples from $\mathbb{H}(A)$ in a PHP admits a Poisson distribution with parameter $\lambda_A := \int_{\mathbb{H}(A)} d\nu$ where ν is the intensity measure of the PHP. Two points $x, y \in \mathbb{R}^d$ will form a hash collision if no hyperplanes cross the interval I_{xy} that connects them. This occurs with probability $\Pr[\text{Poisson}(\lambda_{I_{xy}}) = 0] = \exp(-\lambda_{I_{xy}})$. Thus, if we could choose the PHP intensity ν such that $\lambda_{I_{xy}} = \|x - y\|_p$, we would be done.

For the 2-exponential kernel it now seems natural to sample hyperplanes with uniformly random directions—e.g., via i.i.d. Gaussian entries—and a uniformly random shift from the origin. For $p < 2$, the natural generalization is to replace the Gaussian entries with i.i.d. p-stable entries. Indeed, these choices lead to the desired identity $\lambda_{I_{xy}} = \|x - y\|_p$. Unfortunately, the resulting PHP is not computationally realizable, since $\mathbb{R}^d$ itself is not compact. Simulating the PHP would mean sampling an infinite number of random hyperplanes. Thus, it does not yield a computable LSH scheme.

Our solution to this issue is to restrict the LSH to a compact region C and develop an efficient sampling scheme for the restriction of the PHP to C. At first blush, it may seem like one could simply sample the hyperplane directions with the same p-stable laws and just restrict the random shift to be within the boundaries of C. However, this approach fails, as it skews the PHP intensity measure and

breaks the key identity $\lambda_{I_{xy}} = \|x - y\|_p$. Rather, one must identify the actual marginal intensity on C and devise a method to efficiently sample from it.

Our second main technical step is to show that this is possible for the hypercube $C = \mathcal{Q} := [-1, 1]^d$. We show that the marginal PHP intensity on $\mathcal{Q}$ takes the form of an ℓ_1-*biased* p-*stable law*. We proceed to show how to sample from it efficiently. For $p = 2$, sampling a hyperplane from this law amounts to replacing a single (uniformly random) coordinate in the Gaussian vector with a sample from a two-sided variant of the Rayleigh distribution. For $p < 2$, we do not obtain a closed-form sampling procedure, but we are able to present an efficient rejection sampling scheme. This enables us to simulate the desired PHP on $\mathcal{Q}$ and obtain the kernel LSH schemes that prove Theorem 1.2.

2 The Mosaic LSH for p-Exponential Kernels

2.1 Preliminaries

Poisson hyperplane processes. We introduce notation and review some basic properties of PHPs. We refer to the textbook [42] for further background and terminology.

Let $\mathbb{H}(A)$ be the set of all affine hyperplanes intersecting $A \subset \mathbb{R}^d$. We let

$$H_{z,t} = \{x \in \mathbb{R}^d : z^T x = t\}$$

denote the hyperplane with normal direction $z \in \mathbb{R}^d$ and shift $t \in \mathbb{R}$ from the origin. It will be convenient to allow z to be unnormalized (thus z, t determine $H_{z,t}$ up to multiplication by a scalar).

A PHP is a Poisson point process [27, 30] over $\mathbb{H}(A)$. It is defined by an intensity measure ν over $\mathbb{R}^d \times \mathbb{R}$. For every compact $A \subset \mathbb{R}^d$ with finite non-zero $\nu(\mathbb{H}(A))$, the number $N(A)$ of hyperplanes intersecting A in the PHP is distributed like $\mathrm{Poisson}(\lambda_A)$, where

$$\lambda_A := \int_{\mathbb{H}(A)} d\nu.$$

Conditioned on $N(A)$, the PHP hyperplanes in $\mathbb{H}(A)$ are distributed like i.i.d. samples with density

$$\bar{\nu}^A := \lambda_A^{-1} \cdot \nu|_{\mathbb{H}(A)},$$

which is the normalized restriction of ν to $\mathbb{H}(A)$. This is useful for simulating the PHP.

Lemma 2.1 (PHP simulation)**.** *Suppose we have an algorithm Π_A for sampling from $\bar{\nu}^A$ in time $\mathcal{T}_A$. Then, we can simulate the restriction of a PHP with intensity ν to A in expected time $O(\lambda_A \mathcal{T}_A)$ by sampling $N(A) \sim \mathrm{Poisson}(\lambda_A)$ and then running Π_A for $N(A)$ times.*

p-Stable distributions. We denote by $\mathcal{D}_p$ the distribution with characteristic function $e^{-|t|^p}$. Note that $\mathcal{D}_2$ is the Gaussian distribution $N(0, 2)$. More generally, $\mathcal{D}_p$ is known to exist for $0 < p \leq 2$. It is well-known how to sample from $\mathcal{D}_p$ in time $O(1)$ via the Chambers-Mallows-Stuck (CMS) transform [11], which generalizes the classical Box-Muller transform for Gaussians [43].

Let $\mathcal{D}_p^{\otimes d}$ denote the distribution over $\mathbb{R}^d$ with i.i.d. entries from $\mathcal{D}_p$. If Z is sampled from $\mathcal{D}_p^{\otimes d}$, then $Z^T u$ is distributed like $\|u\|_p \mathcal{D}_p$ for every fixed $u \in \mathbb{R}^d$. This is the p-stable property of $\mathcal{D}_p$.

Let $m_p := \mathbb{E}_{X \sim \mathcal{D}_p} |X|$ denote the first absolute p-stable moment. It is known that m_p is finite for $p \in (1, 2]$ and admits the closed form $m_p = 2\pi^{-1}\Gamma(1 - 1/p)$ [24, Theorem 5.1].

Kernel bandwidth. The p-exponential kernel is $\mathbf{k}(x, y) = \exp(-\|x - y\|_p/\sigma)$ with bandwidth parameter $\sigma > 0$. In Section 1 we let $\sigma = 1$ for simplicity. In this section, it will be more convenient to assume w.l.o.g. that our bounded region is the hypercube $\mathcal{Q} = [-1, 1]^d$ and keep σ explicit. This maintains full generality since Theorem 1.2 can be applied to any bounded region by scaling σ.

2.2 LSH via PHPs

A collection of hyperplanes $\{H_{z^{(j)},t^{(j)}}\}_{j=1}^m$ induces a polygonal space partition called a mosaic. We use the cells of the mosaic to induce a hash function h over $\mathbb{R}^d$. Each point $x \in \mathbb{R}^d$ can be hashed to its cell in time $O(md)$ by computing the length-m bitstring $h(x)_j = \mathbf{1}\{(z^{(j)})^T x - t^{(j)} \geq 0\}$, which encodes whether x is above or below each hyperplane in the collection. See Figure 1 for illustration.

Our LSH schemes will use hyperplanes sampled by simulating appropriate PHPs. For every pair $x, y \in \mathbb{R}^d$, the collision event $h(x) = h(y)$ occurs if the PHP avoids the set $\mathbb{H}(I_{xy})$, where $I_{xy} \subset \mathbb{R}^d$ denotes the interval between x and y. Since $N(I_{xy})$ is distributed like $\text{Poisson}(\lambda_{I_{xy}})$, the avoidance probability is $\exp(-\lambda_{I_{xy}})$. To obtain Theorem 1.2 we need a PHP such that $\lambda_{I_{xy}} = \|x - y\|_p/\sigma$.

We now define the intensity measure ν of our PHP over hyperplanes $H_{Z,T}$. It is the product measure where Z has measure $\frac{1}{\sigma m_p}\mathcal{D}_p^{\otimes d}$ and T has the uniform measure over $\mathbb{R}$. Under this choice of ν, we compute $\lambda_{I_{xy}}$ for a given pair $x, y \in \mathbb{R}^d$ as follows (expectations henceforth are over $Z \sim \mathcal{D}_p^{\otimes d}$):

$$\begin{aligned}
\lambda_{I_{xy}} &= \int_{\mathbb{H}(I_{xy})} d\nu \\
&= \frac{1}{\sigma m_p}\mathbb{E}\left[\int_{\min\{Z^Tx, Z^Ty\}}^{\max\{Z^Tx, Z^Ty\}} dt\right] \\
&= \frac{1}{\sigma m_p}\mathbb{E}\left|Z^Tx - Z^Ty\right| \\
&= \frac{1}{\sigma m_p}\mathbb{E}|Z^T(x-y)|.
\end{aligned}$$

Recall that the p-stable property implies that $Z^T(x-y)$ is distributed like $\|x-y\|_p\mathcal{D}_p$, and hence

$$\mathbb{E}[|Z^T(x-y)|] = \|x-y\|_p\mathbb{E}_{X\sim\mathcal{D}_p}|X| = \|x-y\|_p m_p.$$

Therefore, $\lambda_{I_{xy}} = \|x-y\|_p/\sigma$ as desired.

2.3 Restriction to the Hypercube

The PHP described above cannot be simulated in finite time over $\mathbb{R}^d$ since $\nu(\mathbb{H}(\mathbb{R}^d))$ is not finite. We now show how to simulate it over the hypercube $\mathcal{Q} = [-1, 1]^d$ using Lemma 2.1. That lemma requires *(i)* calculating $\lambda_{\mathcal{Q}}$, and *(ii)* developing a sampling algorithm to draw a sample from $\bar{\nu}^{\mathcal{Q}}$.

We start with $\lambda_{\mathcal{Q}}$. Observe that for every fixed $z \in \mathbb{R}^d$, the inner product z^Tx is maximized over $x \in \mathcal{Q}$ where all coordinates of x have absolute value 1 and their signs match the respective coordinates of z. The same holds for minimizing z^Tx but with the signs of x flipped. Therefore,

$$l_z(\mathcal{Q}) := \min_{x\in\mathcal{Q}} z^Tx = -\sum_{i=1}^d |z_i| = -\|z\|_1 \quad \text{and} \quad u_z(\mathcal{Q}) := \max_{x\in\mathcal{Q}} z^Tx = \sum_{i=1}^d |z_i| = \|z\|_1. \tag{2}$$

Therefore:

$$\lambda_{\mathcal{Q}} = \int_{\mathbb{H}(\mathcal{Q})} d\nu = \frac{1}{\sigma m_p}\mathbb{E}\left[\int_{l_Z(\mathcal{Q})}^{u_Z(\mathcal{Q})} dt\right] = \frac{1}{\sigma m_p}\mathbb{E}\left[2\sum_{i=1}^d |Z_i|\right] = \frac{2}{\sigma m_p}\sum_{i=1}^d \mathbb{E}|Z_i| = \frac{2d}{\sigma}, \tag{3}$$

where we have used that each Z_i has distribution $\mathcal{D}_p$, and is independent of the other coordinates $\{Z_j : j \neq i\}$ of Z, and therefore $\mathbb{E}|Z_i| = m_p$. Plugging Equation (3) in the simulation lemma Lemma 2.1, we find that simulating the PHP from Section 2.2 would take expected time $O(\mathcal{T}_{\mathcal{Q}} \cdot d/\sigma)$ if we had an algorithm for sampling from $\bar{\nu}^{\mathcal{Q}}$ in time $\mathcal{T}_{\mathcal{Q}}$.

Next, we show how to sample from $\bar{\nu}^{\mathcal{Q}}$, which we recall is the probability density proportional to the restriction of ν to $\mathcal{Q}$. We denote by $\bar{\nu}^{\mathcal{Q}}_Z(z)$ the Z-marginal density of $\bar{\nu}^{\mathcal{Q}}$ and by $\bar{\nu}^{\mathcal{Q}}_{T|Z}(t|z)$ the density of T in $\mathcal{Q}$ conditioned on Z. Thus we have $\bar{\nu}^{\mathcal{Q}}(t, z) = \bar{\nu}^{\mathcal{Q}}_{T|Z}(t|z) \cdot \bar{\nu}^{\mathcal{Q}}_Z(z)$. Observe that $\bar{\nu}^{\mathcal{Q}}_{T|Z}(t|z)$ is simply the uniform measure over shifts $t \in \mathbb{R}$ such that $H_{z,t}$ intersects $\mathcal{Q}$. Therefore, we can sample from $\bar{\nu}^{\mathcal{Q}}$ by first sampling z from $\bar{\nu}^{\mathcal{Q}}_Z$ and then sampling t uniformly from $[l_Z(\mathcal{Q}), u_Z(\mathcal{Q})]$.

It thus remains to show how to draw a sample from $\bar{\nu}^{\mathcal{Q}}_Z$. For each fixed $z \in \mathbb{R}^d$ we have

$$\bar{\nu}^{\mathcal{Q}}_Z(z) = \int_{-\infty}^{\infty} d\bar{\nu}^{\mathcal{Q}}(z, t) = \frac{1}{\lambda_{\mathcal{Q}}} \cdot \frac{1}{\sigma m_p} \cdot \mathcal{D}_p^{\otimes d}(z) \int_{l_z(\mathcal{Q})}^{u_z(\mathcal{Q})} dt = \frac{\sigma}{2d} \cdot \frac{1}{\sigma m_p} \cdot \mathcal{D}_p^{\otimes d}(z) \cdot 2\|z\|_1, \tag{4}$$

where we have plugged (2) and (3) for the final equality. We see that the probability density $\bar{\nu}^{\mathcal{Q}}_Z(z)$ equals $\frac{1}{dm_p}\|z\|_1\mathcal{D}_p^{\otimes d}$. This is proportional to $\|z\|_1\mathcal{D}_p^{\otimes d}$, i.e., it is an ℓ_1-*biased p-stable distribution*.

This is an instance of a size-biased distribution, a familiar topic in probability theory. We refer to [4] for background on size-biased distributions.

In summary, we proved the following lemma:

Lemma 2.2. *Suppose we have an algorithm for sampling $z \in \mathbb{R}^d$ with density $\frac{1}{dm_p}\|z\|_1 \mathcal{D}_p^{\otimes d}$ in time $\mathcal{T}$. Then we can simulate in expected time $O(\mathcal{T}d/\sigma)$ a PHP over $\mathcal{Q}$ where the avoidance probability of $\mathbb{H}(I_{xy})$ is $\exp(-\|x-y\|_p/\sigma)$ for every $x, y \in \mathcal{Q}$.*

As a consequence, we get:

Corollary 2.3. *Suppose we have an algorithm for sampling $z \in \mathbb{R}^d$ with density $\frac{1}{dm_p}\|z\|_1 \mathcal{D}_p^{\otimes d}$ in time $\mathcal{T}$. Then, we have an LSH scheme over $\mathcal{Q}$ with sampling time $O(\mathcal{T}d/\sigma)$, evaluation time $O(d^2/\sigma)$, and collision probability $\exp(-\|x-y\|_p/\sigma)$.*

Proof. Section 2.2 shows that the PHP from Lemma 2.2 induces an LSH scheme with the desired collision probability. Sampling a hash function from the LSH scheme means simulating the PHP. Using Lemma 2.1, the simulation time is $O(N\mathcal{T})$ where N is the number of sampled hyperplanes. Since N is distributed like $\mathrm{Poisson}(2d/\sigma)$, the expected LSH sampling time is $O(\mathcal{T}d/\sigma)$. The time to evaluate a sampled LSH map h on a point $x \in \mathbb{R}^d$ is the time it takes to compute the inner product of x with the normal of each hyperplane. Each inner product takes time $O(d)$, and since the number of hyperplanes N is in expectation $2d/\sigma$, the total expected time to evaluate h on x is $O(d^2/\sigma)$. □

2.4 Sampling ℓ_1-Biased p-Stable Vectors

Our goal in this section is to develop a method to sample a vector $z \in \mathbb{R}^d$ with density proportional to $\|z\|_1 \mathcal{D}_p(z)$. By Corollary 2.3, this will complete the sampling algorithm for $\bar{\nu}^{\mathcal{Q}}$, which will allow us to simulate the PHP on $\mathcal{Q}$ with Lemma 2.1, thus proving Theorem 1.2.

Formally, this section proves the following lemma.

Lemma 2.4. *For every $p \in (1, 2]$ there is an algorithm for sampling $z \in \mathbb{R}^d$ with density $\frac{1}{dm_p}\|z\|_1 \mathcal{D}_p^{\otimes d}$ in time $O(d)$ plus expected time $O(1)$. For $p = 2$, the time is just $O(d)$.*

To start, we observe that the additivity of the ℓ_1 norm is useful for reducing the problem of sampling according to the ℓ_1-biased law from sampling a vector to sampling a single coordinate.

Lemma 2.5. *We can sample $z \in \mathbb{R}^d$ with density $\frac{1}{dm_p}\|z\|_1 \mathcal{D}_p(z)$ by the following process:*

1. *Sample I uniformly from $\{1, \ldots, d\}$.*
2. *For all $j \neq I$, sample Z_j from $\mathcal{D}_p$ (scalar p-stable).*
3. *Sample Z_I with density $m_p^{-1}|z|\mathcal{D}_p(z)$ (scalar magnitude-biased p-stable).*
4. *Return the sampled vector $(Z_1, \ldots, Z_d)$.*

Proof. We calculate the density $p_Z(z)$ of $z \in \mathbb{R}^d$ sampled with the above sampling process:

$$
\begin{aligned}
p_Z(z) &= \sum_{i=1}^{d} \Pr[I = i] \cdot p_{Z|I}(z|i) && \text{total probability formula} \\
&= \sum_{i=1}^{d} \frac{1}{d} \cdot \prod_{j=1}^{d} p_{Z_j|I}(z|i) && I \text{ is uniformly random} \\
&= \sum_{i=1}^{d} \frac{1}{d} \left(\frac{|z_i|}{m_p} \mathcal{D}_p(z_i) \cdot \prod_{j \neq i} \mathcal{D}_p(z_j) \right) && Z_i \sim \frac{|z_i|}{m_p}\mathcal{D}_p, \text{ and } Z_j \sim \mathcal{D}_p \text{ for } j \neq i \\
&= \frac{1}{dm_p} \mathcal{D}_p^{\otimes d}(z) \sum_{i=1}^{d} |z_i| && \text{rearranging} \\
&= \frac{1}{dm_p} \|z\|_1 \mathcal{D}_p^{\otimes d}(z) && \text{rearranging}
\end{aligned}
$$

as needed. □

At this point, to complete the sampling algorithm for Lemma 2.4, it remains to show how to sample a scalar z from the magnitude-biased p-stable density $m_p^{-1}|z|\mathcal{D}_p(z)$ (step 3 in Lemma 2.5).

The $p = 2$ case. In this case, we can get a closed-form sampling solution. Recall that the 2-stable distribution is $N(0, 2)$ and its density is $\frac{1}{2\sqrt{\pi}}e^{-z^2/4}$. The corresponding magnitude-biased density is $\frac{1}{4}|z|e^{-z^2/4}$. It is closely related to the Rayleigh distribution with scale $\sqrt{2}$, which is supported on $[0, \infty)$ with density $\frac{1}{2}ze^{-z^2/4}$. The cumulative density function (CDF) of $\text{Rayleigh}(\sqrt{2})$ is $1 - e^{-z^2/4}$, and therefore we can sample $R \sim \text{Rayleigh}(\sqrt{2})$ by drawing $U \sim (0, 1)$ uniformly at random and returning $R = 2\sqrt{-\ln(1 - U)}$. Since we need density $\frac{1}{4}|z|e^{-z^2/4}$ over $\mathbb{R}$ rather than $\frac{1}{2}ze^{-z^2/4}$ over $[0, \infty)$, we can sample a random sign $S \sim \text{Uniform}\{\pm 1\}$ and return the "two-sided Rayleigh" sample SR. This is a sample from the magnitude-biased 2-stable density, $m_2^{-1}|z|\mathcal{D}_2(z)$.

The $1 < p < 2$ case. For this case, we recall the CMS transform [11], which gives a method for sampling p-stables by expressing $\mathcal{D}_p$ as the push-forward measure of closed-form distributions. Let

$$F_p(\theta) := \frac{\sin(p\theta)}{(\cos\theta)^{1/p} \cdot \cos((p-1)\theta)^{(p-1)/p}} \quad \text{and} \quad T_p(\theta, w) := F_p(\theta) \cdot w^{(p-1)/p}.$$

Lemma 2.6 (CMS transform for p-stables [11])**.** *Let μ be the product distribution* $\text{Uniform}(-\frac{\pi}{2}, \frac{\pi}{2}) \times \text{Exp}(1)$ *and let $(\Theta, W) \sim \mu$. Then $Z = T_p(\Theta, W)$ is a sample from $\mathcal{D}_p$.*

By Lemma 2.6 and a standard change of variables for push-forward measures, we get the following:

Corollary 2.7. *Let $\bar{\mu}(\bar{\theta}, \bar{w})$ be the probability density proportional to $|T_p(\bar{\theta}, \bar{w})| \cdot \mu(\bar{\theta}, \bar{w})$. If $(\bar{\Theta}, \bar{W})$ are sampled from $\bar{\mu}$, then $\bar{Z} = T_p(\bar{\Theta}, \bar{W})$ has density proportional to $|z| \cdot \mathcal{D}_p(z)$.*

Both μ and $|T_p(\cdot, \cdot)|$ decompose over their two coordinates, hence $\bar{\mu} \propto |T_p| \cdot \mu$ is a product measure on $\bar{\Theta}$ and $\bar{W}$, so we can sample $\bar{\Theta}$ and $\bar{W}$ independently. Since the W-marginal of μ is $\text{Exp}(1)$, whose density is e^{-w}, the $\bar{W}$-marginal of $\bar{\mu}$ has density $w^{(p-1)/p}e^{-w}$, which is just $\text{Gamma}(1 + \frac{p-1}{p}, 1)$.

For $\bar{\Theta}$, while we do not have a closed form for the density, we can derive an efficient rejection sampling scheme. Recall that in rejection sampling we have a target density $f_{\bar{\Theta}}(\theta)$, a proposal density $g_V(\theta)$, and an envelope bound $m > 0$ that satisfies $f_{\bar{\Theta}}(\theta) \le m \cdot g_V(\theta)$ for all θ in the support of $\bar{\Theta}$. We iteratively sample V with density g_V, accept the sample with probability $f_{\bar{\Theta}}(V)/(m \cdot g_V(V))$, and otherwise we reject the sample and repeat. An accepted sample is distributed with density $f_{\bar{\Theta}}(\theta)$, and the expected number of samples until a sample is accepted is at most m.

Recall that $Y \sim \text{Beta}(\beta, 1)$ is supported on $(0, 1)$ with density $\beta y^{\beta-1}$. Our rejection sampling scheme for $\bar{\Theta}$ is specified in the following lemma. The proof is in Appendix A.

Lemma 2.8. *Let V be a random variable sampled as $V = S \cdot \arccos(Y)$, where $Y \sim \text{Beta}(1-1/p, 1)$ and $S \in \{1, -1\}$ is a uniformly random sign. Then, rejection sampling with V as the proposal for $\bar{\Theta}$ has envelope bound $m = p \le 2$. Therefore, a sample of $\bar{\Theta}$ is returned in expected time $p = O(1)$.*

In summary, by Corollary 2.7, we can sample a scalar magnitude-biased p-stable $z \in \mathbb{R}$, i.e., with density proportional to $|z|\mathcal{D}_p(z)$, as $z = T_p(\bar{\Theta}, \bar{W})$ where $\bar{W} \sim \text{Gamma}(1 + \frac{p-1}{p}, 1)$ and $\bar{\Theta}$ is sampled with the rejection sampling scheme from Lemma 2.8. Putting everything together, the proof of Lemma 2.4 is complete: Putting everything together, the proof of Lemma 2.4 is complete:

Proof of Lemma 2.4. Sample $I \in \{1, \ldots, d\}$ uniformly at random. For $p = 2$, Sample $R \sim \text{Rayleigh}(\sqrt{2})$ and $S \sim \{1, -1\}$. Set $z_I = SR$, and sample the other coordinates of z iid from the normal distribution $N(0, 2)$. The sampling time per coordinate is $O(1)$. For $p \in (1, 2)$, sample $\bar{W} \sim \text{Gamma}(1 + \frac{p-1}{p}, 1)$ and $\bar{\Theta}$ with the rejection sampling scheme from Lemma 2.8, in expected time $O(1)$. Set $z_I = T_p(\bar{\Theta}, \bar{W})$, and sample the other coordinates of z iid from $\mathcal{D}_p$ with the CMS transform (Lemma 2.6) in time $O(1)$ per coordinate. In both cases, per above, z_I has the magnitude-biased p-stable density, and the other coordinates are usual p-stables. Therefore, by Lemma 2.5, the returned vector z has density $\frac{1}{dm_p}\|z\|_1\mathcal{D}_p^{\otimes d}$ as required by Lemma 2.4. □

2.5 The Final Mosaic LSH Scheme

To conclude, we collect the steps to sample a hash function from the mosaic LSH for the p-exponential kernel (with $p \in (1, 2]$):

1. Sample $N \sim \text{Poisson}(2d/\sigma)$.
2. For $j = 1, \ldots, N$:
2.1. Sample $Z^{(j)} \in \mathbb{R}^d$ with the process in Lemma 2.5.
   - For step 2 in Lemma 2.5 (sampling a p-stable):
     - If $p = 2$, sample from $N(0, 2)$.
     - If $p \in (1, 2)$, sample $\Theta \sim [-\frac{\pi}{2}, \frac{\pi}{2}]$ and $W \sim \text{Exp}(1)$ and return $T_p(\Theta, W)$ (Lemma 2.6).
   - For step 3 in Lemma 2.5 (sampling a magnitude-biased p-stable):
     - If $p = 2$, sample $R \sim \text{Rayleigh}(\sqrt{2})$ and a uniformly random sign S and return SR.
     - If $p \in (1, 2)$, sample $\bar{\Theta}$ using the rejection sampling scheme in Lemma 2.8 and $\bar{W} \sim \text{Gamma}(1 + \frac{p-1}{p}, 1)$ and return $T_p(\bar{\Theta}, \bar{W})$ (Corollary 2.7).

2.2. Sample $T^{(j)}$ uniformly in $[-\|Z^{(j)}\|_1, \|Z^{(j)}\|_1]$
3. The hyperplanes $\{H_{Z^{(j)}, T^{(j)}} : j = 1, \ldots, N\}$ induce the mosaic LSH hash function.

Proof of Theorem 1.2. By Lemma 2.4, the sampling scheme in step 2.1 above returns an ℓ_1-biased p-stable vector in at most expected time $O(d)$. Plugging this in Corollary 2.3 yields Theorem 1.2. □

3 Applications to Kernel Density Estimation

The KDE function of a kernel $\mathbf{k}$ and a dataset X is $KDE_X(y) = \frac{1}{|X|} \sum_{x \in X} \mathbf{k}(x, y)$. Many efficient methods for KDE in high dimensions are based on kernel LSH schemes. Theorem 1.2 makes these methods available to p-exponential kernels with $p \in (1, 2]$. We highlight two applications.
Differentially private KDE. Differential privacy (DP) is widely considered the gold standard for privacy-preserving computation in machine learning. It formalizes aggregate computations that leak almost no information about any particular point in the dataset. We refer to [19, 20] and to Appendix B for definitions and background.

Coleman and Shrivastava [17] and Wagner et al. [48] proved that kernel LSH schemes imply efficient DP KDE mechanisms in the function release model. In this model, a privacy-preserving representation of the KDE function can be released and queried infinitely many times without further privacy leakage. Their approach enabled DP KDE mechanisms for the 1-exponential kernel through its LSHability, and, by extension, also for the 2-exponential kernel via an efficient $\ell_2 \to \ell_1$ embedding. Backurs et al. [7] gave a different DP KDE mechanism for the 2-exponential kernel via a more complex reduction to the Gaussian kernel, which in turn was handled in [48] with a different technique.

Theorem 1.2 produces the first explicit DP KDE mechanisms for p-exponential kernels with $p \in (1, 2)$, by plugging it into the LSHability framework of [48] for DP KDE. See Appendix B for details.

Theorem 3.1. *Let $\epsilon, \sigma, d > 0$, let $\alpha, \eta \in (0, 1)$, and let $p \in (1, 2)$. For every set $X \subset \mathcal{Q} \subset \mathbb{R}^d$ that satisfies $|X| \geq \Omega(\log(1/\eta)/(\alpha^2\epsilon))$, the kernel $\mathbf{k}(x, y) = \exp(-\|x - y\|_p/\sigma)$ in $\mathcal{Q}$ admits an ϵ-DP KDE mechanism in the function release model for the KDE function of X, with additive error α, failure probability η, and expected query time $O(d^2 \log(1/\eta)/(\alpha^2\sigma))$.*

KDE with relative error via hashing-based estimators (HBE). Charikar and Siminelakis [13] studied the following form of KDE with relative error. Given $\varepsilon, \mu \in (0, 1)$, the goal is to build a data structure that for each y such that $KDE_X(y) \geq \mu$, it returns an estimate of $KDE_X(y)$ within $(1 \pm \varepsilon)KDE_X(y)$ with high probability. This is useful for small KDE values $\mu \leq KDE_X(y) \ll \varepsilon$ where an additive error of $\pm\epsilon$ is considered too large. By standard concentration, sampling $O(1/\varepsilon^2\mu)$ uniformly random points from X and evaluating the KDE only on them yields a data structure for KDE with this relative error guarantee. Remarkably, [13] showed this can be achieved with a much smaller number of samples, $1/\varepsilon^2\mu^\alpha$ with $\alpha < 1$. Their HBE method achieves $\alpha = 0.5$ for LSHable kernels (see also Algorithm 1 in [6]). Plugging Theorem 1.2 into the HBE, we obtain this result for p-exponential kernels with $1 < p < 2$ (the cases $p = 1, 2$ were already possible with HBE).

Theorem 3.2. *Let $p \in (1, 2)$. The HBE method [13, 6] with the kernel LSH schemes from Theorem 1.2 yields a data structure for KDE with relative error which uses $O(1/\varepsilon^2\sqrt{\mu})$ samples ($\alpha = 0.5$).*

Later work by Charikar et al. [12] showed that KDE with relative error can also be obtained from metric LSH rather than kernel LSH, with even better asymptotic bounds. For the 2-exponential kernel, [12] obtained $\alpha = 0.1$. They did not explicitly handle $p < 2$. However, plugging the classical metric LSH schemes for ℓ_p metrics [18] into their framework yields data structures with $\alpha = 0.25$ for p-exponential kernels with $p \in [1, 2)$. As a result, Theorem 1.2 does not lead to state-of-the-art asymptotic bounds for relative-error KDE. Nevertheless, the downside of [12] is that the method involves substantial combinatorial complexity and seems hard to implement. In contrast, the HBE method has been implemented and shown to be practical in several works [44, 6, 5]. Therefore, Theorem 1.2 yields the first *practical and provable* method for KDE with a relative-error guarantee for $p \in (1, 2)$. The next section includes empirical evaluations for Theorems 3.1 and 3.2.

4 Experimental Evaluation

In this section, we include experimental evaluations to validate our theoretical results. For the values of p we use $p \in \{1.25, 1.5, 1.75, 2\}$. We run experiments on two standard benchmark KDE datasets: Covertype [9] and U.S. Census [34]. Both have been used in prior work on LSH-based methods for KDE [44, 6, 5, 48]. Bandwidth values and other implementation details are specified in Appendix D for reproducibility. Due to space limits, in this section we present the results for Covertype. The results for Census are qualitatively similar and are in Appendix D. Our code is available online at `https://github.com/gorobarak/pExponentialLSH`.

(i) Kernel estimation. Our first experiment validates eq. (1) directly by comparing the empirical collision probability of the mosaic LSH to the true kernel value. The results in Figure 2 show that the error decays quickly as the number of LSH functions sampled increases.

(ii) Differentially private KDE. We implement the LSH-based mechanism for DP-KDE from [48]. The mechanism uses a kernel LSH scheme to generate random features for the data points and then adds privacy-preserving noise to their average. We implement it with the mosaic LSH schemes from Theorem 1.2 and with 100 random features. For $p = 2$, we include as a baseline the same mechanism with the approximate LSH for the 2-exponential kernel that arises by applying an $\ell_2 \to \ell_1$ JL embedding and then using the 1-exponential kernel LSH scheme from [6]. For the other values of p, our results yield the first explicit DP KDE mechanisms, so there is no prior baseline for comparison.

The results in Figure 3 show that for all values of p, the KDE error decays as the privacy budget ε increases. For $p = 2$, our exact kernel LSH and the JL-based approximate LSH perform very similarly (note that DP KDE is inherently approximate due to its built-in privacy-preserving noise).

(iii) KDE with relative error via HBE. We evaluate the HBE method [14, 44, 6] with our mosaic LSH schemes compared to a uniformly random sampling baseline, which, as mentioned in Section 3, achieves the same relative error guarantee for KDE with a larger asymptotic sample size. Figure 4 shows that the HBE method achieves lower error per number of kernel evaluations. The same empirical results were shown in [44, 6] only for the 1-exponential and 2-exponential kernels.

5 Concluding Remarks and Open Questions

We proved exact kernel LSH schemes for p-exponential kernels in bounded domains, including in the important case $p = 2$ which was previously open. This substantially expands the available kernel LSH results in the literature. As applications, we obtained efficient LSHability-based KDE methods.

Our techniques depart from previous LSHability techniques which were based on randomly shifted grids. Rather, we obtain our results by introducing the use of Poisson hyperplane processes into LSH. As such, our LSH schemes are not generalizations of the existing LSH schemes for $p = 1$. In fact, our techniques work for all $p \in (1, 2]$ but *not* for $p = 1$. The technical reason is that the 1-stable distribution (Cauchy) has no finite absolute first moment m_p, which our method requires.

Two notable limitations of our LSH schemes are the $O(d^2)$ dependence in their running time and their restriction to bounded regions. In comparison, existing LSH schemes for $p = 1$ have running time $O(d)$ and they apply to all of $\mathbb{R}^d$, owing to the axis-alignment and translation-invariance of uniform grids in $\mathbb{R}^d$. Our ℓ_1-biased p-stable PHPs do not possess those properties. Whether these aspects of Theorem 1.2 can be improved, particularly for $p = 2$, is an interesting open problem.

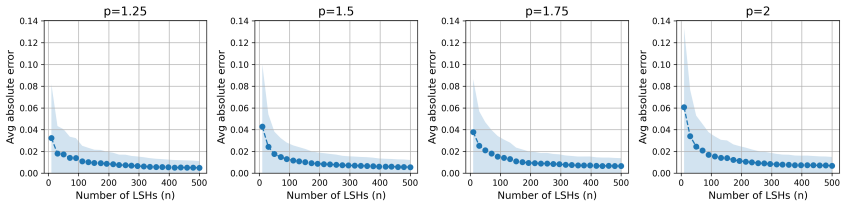


Figure 2: p-Exponential kernel estimation by mosaic LSH on the Covertype dataset

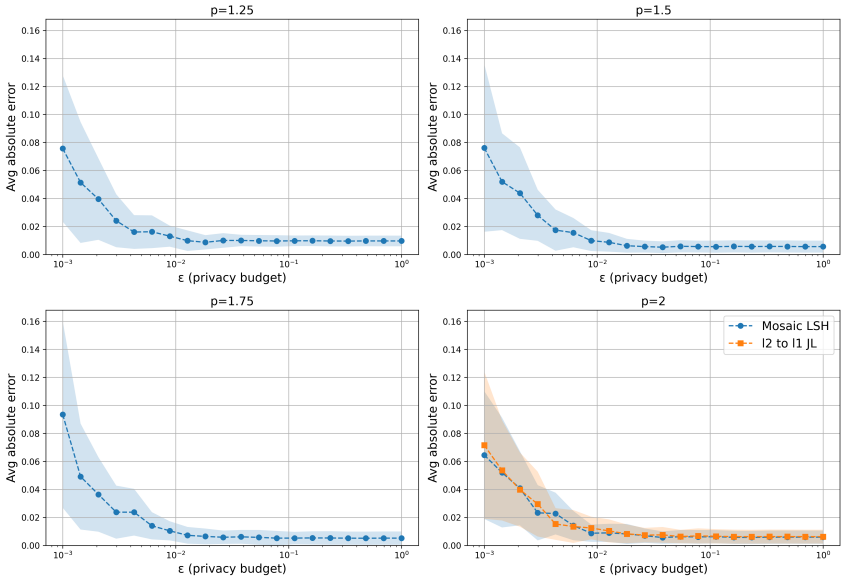


Figure 3: Differentially private KDE results for Covertype

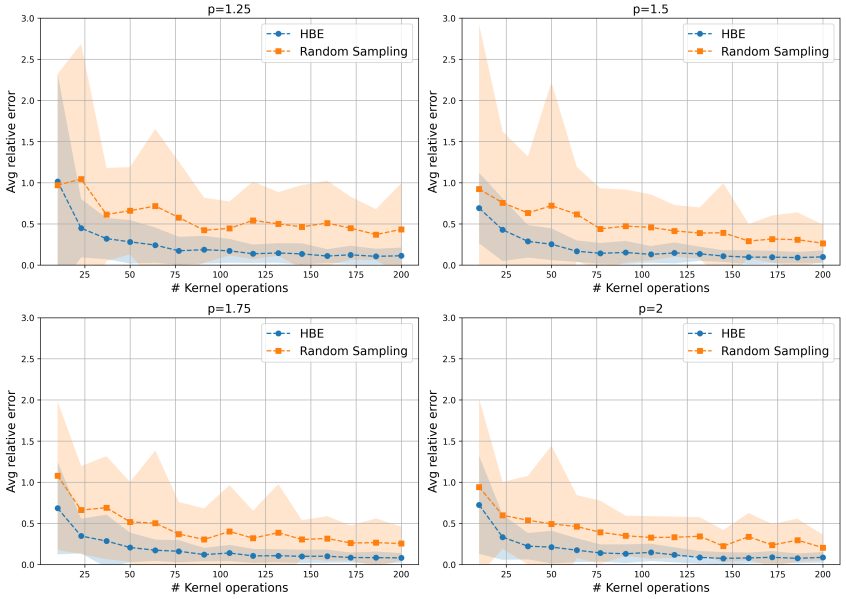


Figure 4: KDE via HBE results for Covertype

AI Use Statement

During the course of this research, OpenAI's GPT 5.0 through 5.4 were used to assist in literature search, understanding mathematical background and exploring proof ideas. The text and the proofs in the paper were written without AI assistance. Anthropic's Claude Opus 4.7 was used for interactive coding for the experiments. The core part of the code (the implementation of mosaic LSH) was written without AI assistance, and the code base as a whole was thoroughly verified. The authors are accountable for all of the text, proofs and artifacts associated with this paper.

Acknowledgments and Disclosure of Funding

This research was supported by the Israeli Ministry of Innovation, Science & Technology, by Len Blavatnik and the Blavatnik Family foundation, and by an Alon Scholarship.

References

[1] Nir Ailon and Bernard Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on computing*, 39(1):302–322, 2009.

[2] Alex Andoni and Piotr Indyk. Dimension reduction in kernel spaces from locality-sensitive hashing. 2009.

[3] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.

[4] Richard Arratia, Larry Goldstein, and Fred Kochman. Size bias for one and all. *Probability Surveys*, 16:1–61, 2019.

[5] Arturs Backurs, Piotr Indyk, Cameron Musco, and Tal Wagner. Faster kernel matrix algebra via density estimation. In *International Conference on Machine Learning*, pages 500–510. PMLR, 2021.

[6] Arturs Backurs, Piotr Indyk, and Tal Wagner. Space and time efficient kernel density estimation in high dimensions. *Advances in neural information processing systems*, 32, 2019.

[7] Arturs Backurs, Zinan Lin, Sepideh Mahabadi, Sandeep Silwal, and Jakub Tarnawski. Efficiently computing similarities to private datasets. *arXiv preprint arXiv:2403.08917*, 2024.

[8] Matej Balog, Balaji Lakshminarayanan, Zoubin Ghahramani, Daniel M Roy, and Yee Whye Teh. The mondrian kernel. *arXiv preprint arXiv:1606.05241*, 2016.

[9] Jock Blackard. Covertype. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C50K5N.

[10] Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997.

[11] John M Chambers, Colin L Mallows, and BW Stuck. A method for simulating stable random variables. *Journal of the american statistical association*, 71(354):340–344, 1976.

[12] Moses Charikar, Michael Kapralov, Navid Nouri, and Paris Siminelakis. Kernel density estimation through density constrained near neighbor search. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 172–183. IEEE, 2020.

[13] Moses Charikar and Paris Siminelakis. Hashing-based-estimators for kernel density in high dimensions. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1032–1043. IEEE, 2017.

[14] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.

[15] Flavio Chierichetti and Ravi Kumar. Lsh-preserving functions and their applications. *Journal of the ACM (JACM)*, 62(5):1–25, 2015.

[16] Benjamin Coleman and Anshumali Shrivastava. Sub-linear race sketches for approximate kernel density estimation on streaming data. In *Proceedings of The Web Conference 2020*, pages 1739–1749, 2020.

[17] Benjamin Coleman and Anshumali Shrivastava. A one-pass distributed and private sketch for kernel sums with applications to machine learning at scale. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3252–3265, 2021.

[18] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004.

[19] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[20] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3-4):211–487, 2014.

[21] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning1. *The Annals of Statistics*, 36(3):1171–1220, 2008.

[22] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 189–197. IEEE, 2000.

[23] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.

[24] Svante Janson. Stable distributions. *arXiv preprint arXiv:1112.0220*, 2011.

[25] William B Johnson and Gideon Schechtman. Embedding lpm into l 1 n. *Acta Mathematica*, 149(1):71–85, 1982.

[26] Sahil Joshi, Agniva Chowdhury, Amar Kanakamedala, Ekam Singh, Evan Tu, and Anshumali Shrivastava. Race attention: A strictly linear-time attention for long-sequence training. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

[27] John Frank Charles Kingman. *Poisson processes*, volume 3. Clarendon Press, 1992.

[28] Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. Lp-norm multiple kernel learning. *The Journal of Machine Learning Research*, 12:953–997, 2011.

[29] Maria Despoina Koltsidopoulou, Andrew Pavlides, Dionissios T Hristopulos, and Emmanouil A Varouchakis. Enhancing geostatistical analysis of natural resources data with complex spatial formations using non-euclidean distances. *Mathematical Geosciences*, pages 1–21, 2026.

[30] Günter Last and Mathew Penrose. *Lectures on the Poisson process*, volume 7. Cambridge University Press, 2018.

[31] Meng Liu and David F Gleich. Strongly local p-norm-cut algorithms for semi-supervised learning and local graph clustering. *Advances in neural information processing systems*, 33:5023–5035, 2020.

[32] Zichang Liu, Zhaozhuo Xu, Benjamin Coleman, and Anshumali Shrivastava. One-pass distribution sketch for measuring data heterogeneity in federated learning. *Advances in Neural Information Processing Systems*, 36:15660–15679, 2023.

[33] Jiří Matoušek. On variants of the johnson–lindenstrauss lemma. *Random Structures & Algorithms*, 33(2):142–156, 2008.

[34] Thiesson Bo Meek, Chris and David Heckerman. US Census Data (1990). UCI Machine Learning Repository, 2001. DOI: https://doi.org/10.24432/C5VP42.

[35] Rajeev Motwani, Assaf Naor, and Rina Panigrahi. Lower bounds on locality sensitive hashing. In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 154–157, 2006.

[36] Huy L Nguyen. Approximate nearest neighbor search in ℓ_p. *arXiv preprint arXiv:1306.3601*, 2013.

[37] Eliza O'Reilly and Ngoc Mai Tran. Stochastic geometry to generalize the mondrian process. *SIAM Journal on Mathematics of Data Science*, 4(2):531–552, 2022.

[38] Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *ACM Transactions on Computation Theory (TOCT)*, 6(1):1–13, 2014.

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

[40] Daniel M Roy and Yee W Teh. The mondrian process. In *Advances in neural information processing systems*, pages 1377–1384, 2008.

[41] Saverio Salzo, Lorenzo Rosasco, and Johan Suykens. Solving lp-norm regularization with tensor kernels. In *International Conference on Artificial Intelligence and Statistics*, pages 1655–1663. PMLR, 2018.

[42] Rolf Schneider and Wolfgang Weil. *Stochastic and integral geometry*. Springer, 2008.

[43] David W Scott. Box–muller transformation. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3(2):177–179, 2011.

[44] Paris Siminelakis, Kexin Rong, Peter Bailis, Moses Charikar, and Philip Levis. Rehashing kernel evaluation in high dimensions. In *International Conference on Machine Learning*, pages 5789–5798. PMLR, 2019.

[45] Ryan Spring and Anshumali Shrivastava. A new unbiased and efficient class of lsh-based samplers and estimators for partition function computation in log-linear models. *arXiv preprint arXiv:1703.05160*, 2017.

[46] Emmanouil A Varouchakis, Maria Despoina Koltsidopoulou, and Andrew Pavlides. Designing robust covariance models for geostatistical applications. *Stochastic Environmental Research and Risk Assessment*, 39(6):2517–2527, 2025.

[47] Tal Wagner. Learning from end user data with shuffled differential privacy over kernel densities. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

[48] Tal Wagner, Yonatan Naamad, and Nina Mishra. Fast private kernel density estimation via locality sensitive quantization. In *International Conference on Machine Learning*, pages 35339–35367. PMLR, 2023.

[49] Daniel Wolfson and Tal Wagner. Positional lsh: Binary block matrix approximation for attention with linear biases. *Advances in neural information processing systems*, 2026.

[50] Amir Zandieh, Navid Nouri, Ameya Velingker, Michael Kapralov, and Ilya Razenshteyn. Scaling up kernel ridge regression via locality sensitive hashing. In *International Conference on Artificial Intelligence and Statistics*, pages 4088–4097. PMLR, 2020.

A Rejection Sampling for $p \in (1,2)$: Proof of Lemma 2.8

Recall that in rejection sampling, we have a target density $f(x)$ we want to sample from and such that we can evaluate it at any given point x, and a proposal density $g(x)$ which we can sample from, such that they satisfy the envelope property $\forall x,\ f(x) \le m \cdot g(x)$ for some $m > 0$. The rejection sampler operates as follows:

1. Sample $X \sim g$ and $U \sim \mathrm{Uniform}(0,1)$.
2. If $f(X) \ge mUg(X)$, accept X as a sample from f and terminate.
3. Otherwise, reject the sample and return to step 1.

An accepted sample is provably distributed according to f. The expected number of iterations until acceptance is at most m.

Our goal is to devise a rejection sampling scheme for $\bar{\Theta}$. Since the Θ-marginal of μ is uniform, and $\bar{\mu}$ is proportional to $|F_p(\theta) w^{(p-1)/p}| \cdot \mu$, the $\bar{\Theta}$-marginal of $\bar{\mu}$ is proportional to $|F_p(\theta)|$. We thus need a rejection sampling scheme for sampling $\theta \sim (-\frac{\pi}{2}, \frac{\pi}{2})$ with target density proportional to $|F_p(\theta)|$.

We start by calculating the normalization factor. Let $G_p(\theta)$ be defined over $\theta \in [0, \frac{\pi}{2})$ as

$$G_p(\theta) = \cos((p-1)\theta)^{1/p} \cos(\theta)^{1-1/p}.$$

One can verify by direct derivation that $\frac{d}{d\theta} G_p(\theta) = -(1-\frac{1}{p}) F_p(\theta)$. Consequently, using the fact that $|F_p(\theta)|$ is symmetric about zero and equals to $F_p(\theta)$ at $\theta \in [0, \frac{\pi}{2})$,

$$\int_{-\pi/2}^{\pi/2} |F_p(\theta)| d\theta = 2 \int_0^{\pi/2} F_p(\theta) d\theta = 2 \cdot \frac{-G_p(\pi/2) + G_p(0)}{1 - 1/p} = \frac{2}{1-1/p}.$$

Therefore, the exact target density we need to sample from is

$$f_{\bar{\Theta}}(\theta) = \tfrac{1}{2} \left(1 - \tfrac{1}{p}\right) |F_p(\theta)|.$$

We now define the proposal density $g_V(v)$ by the following sampling process: Sample $Y \sim \mathrm{Beta}(1 - 1/p, 1)$ and a uniformly random sign $S \in \{1, -1\}$. Let $R = \arccos(Y)$. Return the sample $V = SR$.

We derive a closed form for $g_V(\theta)$. Since $Y \sim \mathrm{Beta}(\beta, 1)$ takes values in $(0,1)$ with density $\beta y^{\beta-1}$, then R takes values in $(0, \frac{1}{2}\pi)$ and its density $f_R(\theta)$ is, by the change of variables formula for densities with $Y = \cos(R)$,

$$f_R(\theta) = \left(1 - \tfrac{1}{p}\right) (\cos(\theta))^{-1/p} \left| \tfrac{d}{d\theta} \cos(\theta) \right| = \left(1 - \tfrac{1}{p}\right) (\cos(\theta))^{-1/p} \sin(\theta).$$

Therefore, after the multiplication by the random sign S, we find that V takes values in $(-\frac{1}{2}\pi, \frac{1}{2}\pi)$ with density

$$g_V(\theta) = \tfrac{1}{2} \left(1 - \tfrac{1}{p}\right) (\cos(\theta))^{-1/p} |\sin(\theta)|.$$

Finally, it remains to calculate the envelope parameter m. Since both the target density $f_{\bar{\Theta}}(\theta)$ and the proposal density $g_V(\theta)$ are symmetric about zero, it suffices to calculate m for $\theta \in (0, \frac{1}{2}\pi)$. In this range, for $p \in (1,2)$ we have $|\sin(p\theta)| = \sin(p\theta)$. Therefore, for all $\theta \in (0, \frac{1}{2}\pi)$,

$$\begin{aligned} m_\theta := \frac{f_{\bar{\Theta}}(\theta)}{g_V(\theta)} &= \frac{\frac{1}{2}\left(1 - \frac{1}{p}\right) (\cos(\theta))^{-1/p} (\cos((1-p)\theta))^{-(1-1/p)} \sin(p\theta)}{\frac{1}{2}\left(1 - \frac{1}{p}\right) (\cos(\theta))^{-1/p} \sin(\theta)} \\ &= \frac{\sin(p\theta)}{\sin(\theta)(\cos((1-p)\theta))^{1-1/p}}. \end{aligned}$$

Let $\theta' = (p-1)\theta$. Then the above becomes

$$\begin{aligned} m_\theta &= \frac{\sin(\theta+\theta')}{\sin(\theta)(\cos(\theta')^{1-1/p}} \\ &\le \frac{\sin(\theta+\theta')}{\sin\theta\cdot\cos\theta'} && (\cos(\theta'))^{1/p} \le 1 \\ &= \frac{\sin\theta\cdot\cos\theta' + \sin\theta'\cdot\cos\theta}{\sin\theta\cdot\cos\theta'} && \text{sum-product trigonometric identity} \\ &= 1+\tan\theta'\cdot\cot\theta. \end{aligned}$$

Since the function $x \mapsto \tan(x)/x$ is increasing on $(0, \frac{1}{2}\pi)$, and $\theta' \le \theta$, we have

$$\tan\theta' \le \frac{\theta'}{\theta}\cdot\tan\theta = (p-1)\tan\theta,$$

and therefore the above yields

$$m_\theta \le 1 + (p-1)\cdot\tan\theta\cdot\cot\theta = p.$$

Therefore, the envelope bound for rejection sampling with target density $f_{\tilde{\Theta}}(\theta)$ and proposal density g_V is $m \le p$, and Lemma 2.8 is proven.

B Omitted Details from Section 3: DP KDE

We recall the differential privacy definitions relevant to Theorem 3.1.

Definition B.1 (differential privacy)**.** *Let $\epsilon > 0$. Let M be a randomized algorithm that maps datasets with elements in $\mathcal{X}$ to a space of outputs $\mathcal{O}$. M is called an ϵ-differentially private (ϵ-DP) mechanism if for every two datasets X, X' such that X' is obtained from X by either adding or dropping a single element, and for every set of outputs $O \subset \mathcal{O}$, it holds that $\Pr[M(X) \in O] \le e^\epsilon \Pr[M(X') \in O]$.*

Definition B.2 (DP KDE in the function release model)**.** *Let $\epsilon > 0$ and $\alpha, \eta \in (0,1)$. Let $\mathbf{k} : \mathcal{X}\times\mathcal{X} \to [0,1]$ be a kernel. In the DP KDE problem in the function release model, the goal is to design an ϵ-DP mechanism M that given a dataset $X \subset \mathcal{X}$, returns a function $F_X : \mathcal{X} \to [0,1]$ that for each $y \in \mathcal{X}$ satisfies $\Pr[|F_X(y) - KDE_X(y)| \le \alpha] \ge 1-\eta$.*

The mechanisms we obtain in Theorem 3.1 satisfy the above definitions. Theorem 3.1 follows from plugging Theorem 1.2 into Theorem D.3 and Proposition D.4 in [48]. The only thing to check is that the efficiency bounds in Proposition D.4 in [48] depend on $\log B$, where B is the number of bins of the LSH scheme. In our case, the number of bins is 2^N where $N \sim \text{Poisson}(2d/\sigma)$ is the number of hyperplanes sampled in the mosaic LSH scheme. Therefore, $\mathbb{E}[\log_2 B] = \mathbb{E}[N] = 2d/\sigma$.

C Non-LSHability of the Gaussian Kernel

In this section we show a negative result, proving the lack of LSH schemes for a class of kernels that includes the Gaussian kernel, which is the $p = 2$ case of the theorem below.

Theorem C.1. *Let $p \ge 1$ and $d \ge 1$. Consider the kernel $\mathbf{k}_p(x,y) = \exp(-\|x-y\|_p^p)$ over $\mathbb{R}^d$. Then, $\mathbf{k}_p$ is LSHable if and only if $p = 1$.*

Proof. The $p = 1$ is the 1-exponential kernel whose LSHability is known. We now assume that $p > 1$ and show that $\mathbf{k}$ is not LSHable.

Charikar [14] showed that if $\mathbf{k} : \mathcal{X}\times\mathcal{X} \to [0,1]$ is LSHable then $1-\mathbf{k}$ satisfies the triangle inequality. Therefore, if $\mathbf{k}_p$ is LSHable then the following must hold $\forall a, b, c \in \mathcal{X}$:

$$\begin{aligned} &1 - \mathbf{k}_p(a,b) + 1 - \mathbf{k}_p(b,c) \ge 1 - \mathbf{k}_p(a,c) \\ &\iff \mathbf{k}_p(a,c) - \mathbf{k}_p(a,b) - \mathbf{k}_p(b,c) + 1 \ge 0. \end{aligned} \tag{5}$$

We construct a contradictory example where eq. (5) fails to hold and this will conclude our proof. It suffices to prove the claim for $d = 1$ since a negative example for $\mathbb{R}$ embeds in $\mathbb{R}^d$ for all $d \ge 1$.

Take $\mathcal{X} = \mathbb{R}$ and $a = 0,\ b = x,\ c = 2x$, as three evenly spaced points where $x \in \mathbb{R}^+$. Then eq. (5) becomes

$$\begin{aligned} &\mathbf{k}_p(a,c) - \mathbf{k}_p(a,b) - \mathbf{k}_p(b,c) + 1 \geq 0 \\ \iff\ & e^{-\|2x\|_p^p} - e^{-\|x\|_p^p} - e^{-\|x\|_p^p} + 1 \geq 0 \\ \iff\ & e^{-(2x)^p} - 2e^{-x^p} + 1 \geq 0. \end{aligned} \tag{6}$$

We denote the left-hand side of eq. (6) by $f_p(x) \coloneqq e^{-(2x)^p} + 1 - 2e^{-x^p}$. We show that there is $x \in \mathbb{R}^+$ such that $f_p(x) < 0$. We apply a change of variable $u = x^p$ and define

$$g(u) = e^{-2^p u} + 1 - 2e^{-u},$$

so we have $g(u) = f_p(x)$. At $u = 0$ we have $g(0) = 0$. We calculate the derivative:

$$g'(u) = -2^p e^{-2^p u} + 2e^{-u}.$$

At $u = 0$,

$$g'(0) = 2e^{-0} - 2^p e^{-0} = 2 - 2^p < 0,$$

where the last inequality is since $p > 1$. Thus we have $g(0) = 0$ and $g'(0) < 0$. Therefore, for sufficiently small $u^* > 0$, we have $g(u^*) < 0$, and thus for $x^* = (u^*)^{\frac{1}{p}} > 0$ we get $f_p(x^*) < 0$. □

D Additional Experimental Details and Results

Dataset parameters. Covertype has 581K points and dimension 55. Census has 2.45M points and dimension 68. For efficiency we sample a subset of each dataset (note that subsampling preserves all KDE values). For Covertype we sample at rate 25% and obtain 145K points. For Census we sample at rate 5% and obtain 123K points.

Bandwidth selection. In line with prior work, we choose two bandwidth settings for each kernel. One is chosen to render typical KDE values of order 10^{-2} for experiments measuring additive error. We use it for the kernel estimation and DP KDE experiments. The other bandwidth setting is chosen to render typical KDE values of order 10^{-3} for experiments measuring relative error. We use this setting for HBE experiments since as explained in Section 3, the method is designed for regimes where KDE values we need to approximate are very small.

For Covertype, the bandwidth settings are 0.15 and 0.05 respectively. For Census, the bandwidth settings are 0.42 and 0.2 respectively. Note that those values account for scaling the coordinates of each dataset into the range $[-1, 1]$ to align with the normalization in Theorem 1.2.

Experimental parameters. The kernel estimation experiments measure average absolute additive error over 1000 random pairs of points. The DP KDE experiments measure the average absolute additive error over 100 query points chosen at random from the dataset. The HBE experiments measure the average relative error over 100 query points chosen at random from the dataset. Error bars ($\pm$std) are shown as shaded regions around each curve.

Additional experimental results. The results for the Census datasets are shown in Figures 5 to 7.

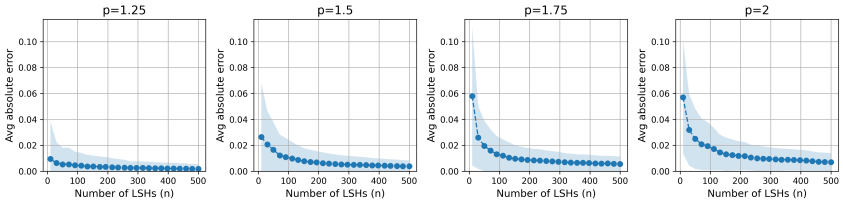


Figure 5: p-Exponential kernel estimation by mosaic LSH on the Census dataset

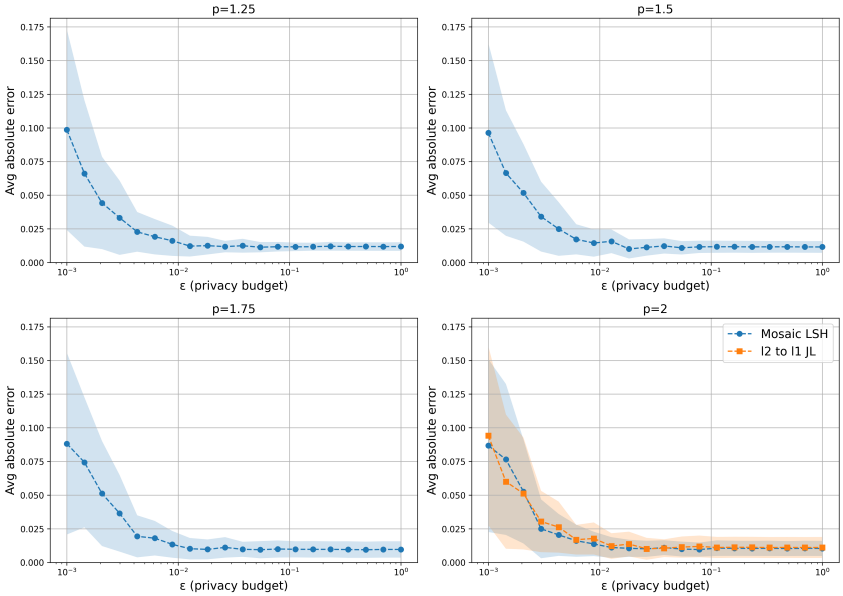


Figure 6: Differentially private KDE results for Census

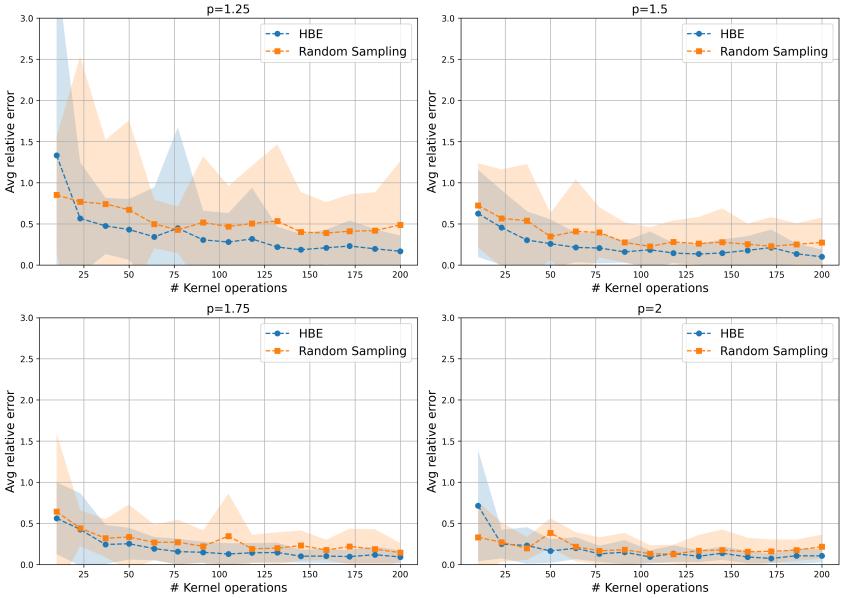


Figure 7: KDE via HBE results for Census

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer:[Yes]

   Justification: he main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations discussed in concluding section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper is mathematical and full rigorous proofs are provided.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experiments are on public benchmark datasets, all parameters are specified, and our code is enclosed with the submission.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Data is public, code is enclosed and will be made public with the release of the paper

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All details specified

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars reported

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: The experiments in the scope of the paper are computationally trivial to run on any machine; no heavy compute is needed

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This papers adheres to the code of ethics

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is a theoretical paper on a fundamental problem without broader impacts that need to be specifically discussed

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This is a theoretical paper on a fundamental problem without a need to specialized safeguards

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets used with accordance to their licenses

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Not relevant to this paper

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Not relevant to this paper

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: No such use of LLM in the research presented in this paper

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.