CS798L PROJECT REPORT

Differential Privacy In Data Reduction

Shlok Mishra
218170991
Statistics and Data Science

Srishti Chandra
221088
Computer Science and Engineering

September 13, 2025

1 Introduction and Motivation

1.1 The Curse of Dimensionality

High–dimensional datasets are ubiquitous in modern data science— ranging from genomic expression profiles to sparse user–item interaction logs. Yet, as dimensionality d grows, geometric and statistical intuitions begin to fail: inter-point distances concentrate, nearest-neighbour search degrades, and sample complexity explodes—phenomena collectively dubbed the *curse of dimensionality* [6]. Figure 1 visualises how pairwise ℓ_2 distances become nearly indistinguishable beyond a few dozen dimensions, eroding the contrast that underpins many learning algorithms.

1.2 Privacy Risks in High-Dimensional Data

The same rich feature spaces that empower expressive models also amplify privacy leakage. Simple record-linkage attacks can re-identify individuals from apparently anonymised micro-data once dimensionality is high [4]. While Differential Privacy (DP) bounds such leakage [4], naively enforcing DP on high-dimensional vectors may require noise of magnitude $\Omega(\sqrt{d})$, obliterating accuracy. This tension motivates transformations that *simultaneously* reduce dimension and temper sensitivity.

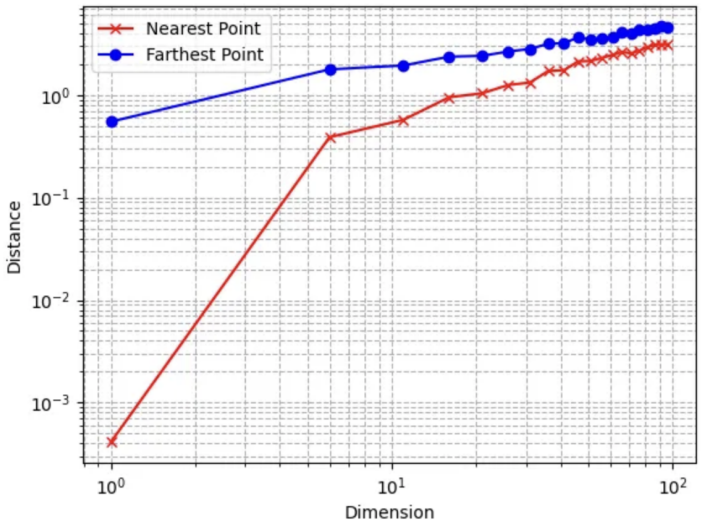
1.3 Our Focus and Contributions

Random projections—specifically the Johnson–Lindenstrauss (JL) transform—form a promising avenue. Two seminal works crystallise the idea:

- **Kenthapadi et al.** show that applying a JL transform *followed by Gaussian noise* yields strong (ε, δ)-DP while preserving pairwise distances up to a small bias [7].
- **Blocki et al.** go further, proving that the JL transform *alone* is sufficiently random to guarantee DP for a broad class of linear queries—most notably graph cuts and covariance estimates [3].

This survey distills and contrasts those results within the broader DP toolbox. Concretely, we:

1. **Synthesize** the privacy proofs for projection-plus-noise and "inherently private" JL schemes, highlighting common structural lemmas.



Distances to nearest and farthest points as n increases (Image by the author)

Figure 1: Illustration of distance concentration in high-dimensional spaces ("curse of dimensionality").

2. **Compare** JL-based mechanisms with classic input/output perturbation and Multiplicative Weights approaches in terms of privacy budget, utility loss, and scalability.

3. **Identify open problems**—e.g. optimal target dimension k under joint privacy–utility constraints and extensions to non-Euclidean metrics.

The remainder of the report is organised as follows: Section 2 reviews the JL lemma and DP preliminaries; Section 3 dissects privacy-preserving projection algorithms; Section 4 benchmarks them against alternatives; Section 5 summarises theoretical bounds; and Section 6 outlines directions for future work.

2 Background

2.1 Notation and Problem Set-up

We write a dataset as a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ whose i-th row $\mathbf{x}_i \in \mathbb{R}^d$ represents one individual's record; n is the number of individuals and $d \gg n$ in our motivating scenarios. Throughout the paper we use:

- $\|\cdot\|_2$ for the Euclidean norm and $\|\cdot\|_1$ for the ℓ_1 norm.
- **Sensitivity.** For a function $f : \mathbb{R}^{n\times d} \to \mathbb{R}^m$ its ℓ_1-*sensitivity* is $S_1(f) = \max_{\text{nbrs } \mathbf{X},\mathbf{X}'} \|f(\mathbf{X}) - f(\mathbf{X}')\|_1$, where "nbrs" means the two datasets differ in exactly one row. ℓ_2-sensitivity $S_2(f)$ is defined analogously with $\|\cdot\|_2$.
- A *random projection* or *JL matrix* $\mathbf{P} \in \mathbb{R}^{k\times d}$, $k \ll d$, with rows drawn i.i.d. either from $\mathcal{N}(0, (1/k)\mathbf{I}_d)$ (Gaussian JL) or from a sparse Rademacher distribution [1].

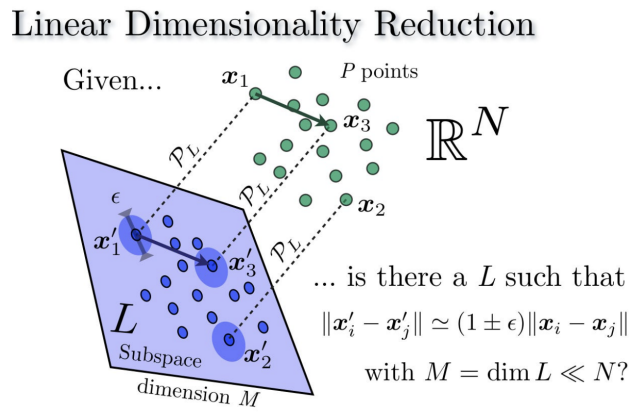


Figure 2: Geometric intuition for the Johnson–Lindenstrauss transform: random projection flattens a point cloud from $\mathbb{R}^N$ into $\mathbb{R}^M$ while keeping pairwise distances within $(1 \pm \varepsilon)$ multiplicative distortion.

2.2 Johnson–Lindenstrauss Transform

Intuitively, the Johnson–Lindenstrauss (JL) lemma asserts that a small *random* subspace is sufficient to preserve pairwise distances, making it a powerful dimensionality-reduction primitive. Figure 2 offers the geometric caricature.

[Johnson–Lindenstrauss [6]] For any $0 < \varepsilon < 1$ and any set V of n points in $\mathbb{R}^d$, there exists a mapping $f : \mathbb{R}^d \to \mathbb{R}^k$ with $k = O\big(\frac{\log n}{\varepsilon^2}\big)$ such that for all $\mathbf{u}, \mathbf{v} \in V$

$$(1-\varepsilon)\,\|\mathbf{u}-\mathbf{v}\|_2^2 \;\le\; \|f(\mathbf{u})-f(\mathbf{v})\|_2^2 \;\le\; (1+\varepsilon)\,\|\mathbf{u}-\mathbf{v}\|_2^2.$$

A random matrix $\mathbf{P} \in \mathbb{R}^{k\times d}$ whose entries $P_{ij} \sim \mathcal{N}(0, 1/k)$ satisfies the lemma with probability at least $1 - \frac{1}{n^3}$.

Practical constructions. Besides dense Gaussian projections, database-friendly variants use Rademacher or very sparse sign matrices to cut time and storage to $O(d \log d)$ or even $O(d)$ [1]. We shall note in Section 3 that these variants inherit essentially the same privacy properties.

2.3 Differential Privacy Primer

[Differential Privacy [4]] A randomized algorithm $\mathcal{A}$ satisfies (ε, δ)-differential privacy (DP) if for every pair of neighboring datasets $\mathbf{X}, \mathbf{X}'$ and for every measurable set $S \subseteq \mathrm{Range}(\mathcal{A})$,

$$\Pr[\mathcal{A}(\mathbf{X}) \in S] \;\le\; e^{\varepsilon}\Pr[\mathcal{A}(\mathbf{X}') \in S] + \delta.$$

Canonical mechanisms. DP is usually enforced by adding noise calibrated to sensitivity:

- **Laplace mechanism** adds i.i.d. $\mathrm{Lap}(S_1(f)/\varepsilon)$ noise to each coordinate of $f(\mathbf{X})$, achieving $(\varepsilon, 0)$-DP.
- **Gaussian mechanism** adds i.i.d. Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma \ge S_2(f)\sqrt{2\ln(1.25/\delta)}/\varepsilon$, providing (ε, δ)-DP and tighter concentration around zero [5].

In high-dimensional settings, the magnitudes $S_1(f), S_2(f)$ may scale with $\sqrt{d}$, hence direct noise addition can overwhelm signal. Sections 3–4 illustrate how a JL projection shrinks sensitivity before noise is introduced—or, in some cases, *eliminates* the need for noise entirely.

3 Privacy-Preserving Dimensionality Reduction

3.1 Projection + Noise: PrivateProjection

The method of Kenthapadi *et al.* [7] first applies a Johnson–Lindenstrauss (JL) projection to shrink sensitivity and then adds Gaussian noise.

Algorithm 1 PrivateProjection (Kenthapadi *et al.*)

Require: Dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$, target dim. k, privacy parameters (ε, δ)

1: Draw $\mathbf{P} \sim \mathcal{N}(0, 1/k)^{k \times d}$
2: $\mathbf{Y} \leftarrow \mathbf{X}\mathbf{P}^\top$ // JL projection
3: $\sigma \leftarrow w_2(\mathbf{P})\sqrt{2\ln(1.25/\delta)}/\varepsilon$
4: $\tilde{\mathbf{Y}} \leftarrow \mathbf{Y} + \mathcal{N}(0, \sigma^2 I_k)^{\otimes n}$
5: **return** $\tilde{\mathbf{Y}}$ =0

[Privacy & Utility of PrivateProjection] Algorithm 1 satisfies (ε, δ)-DP. For any rows $\mathbf{u}, \mathbf{v}$,

$$\left| \|\tilde{\mathbf{u}} - \tilde{\mathbf{v}}\|_2^2 - \|\mathbf{u} - \mathbf{v}\|_2^2 \right| = O\Big(\tfrac{\|\mathbf{u}-\mathbf{v}\|_2^2}{\sqrt{k}} + k\sigma^2 \Big) \quad \text{with high probability.}$$

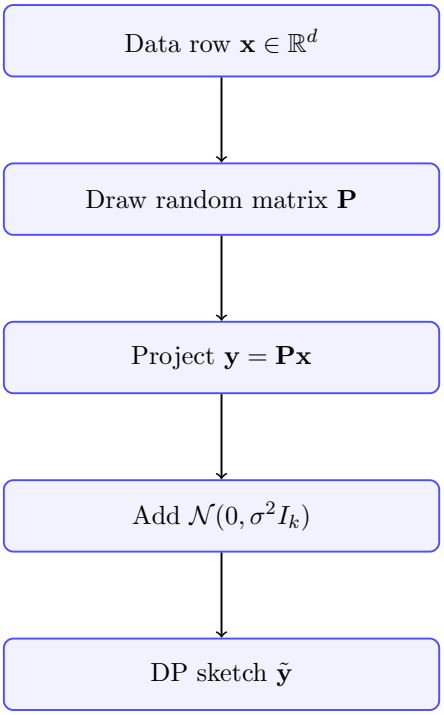


Figure 3: PrivateProjection pipeline.

3.2 Projection-Only: JL-CutSketch

Blocki *et al.* [3] prove that the JL projection *alone* is already differentially private for rank-1 neighbouring datasets (e.g. adding or removing one edge in a graph) once the input is lightly smoothed.

Algorithm 2 JL-CutSketch (Blocki *et al.*)

Require: Graph $G = (V, E)$, smoothing weight w, privacy (ε, δ)

1: Add $w/|E|$ to every edge weight // smooth
2: Let $\mathbf{B} \in \{0, \pm 1\}^{|E| \times |V|}$ be the incidence matrix
3: Draw $\mathbf{P} \sim \mathcal{N}(0, 1/k)^{k \times |E|}$
4: $\mathbf{Y} \leftarrow \mathbf{PB}$
5: Publish $\mathbf{Y}$
6: **return** Function that answers any cut query via $\mathbf{Y}$ =0

[Privacy of JL-CutSketch] With $k = \Theta(\varepsilon^{-2} \log(1/\delta))$, Algorithm 2 satisfies (ε, δ)-DP. Cut queries are answered with multiplicative $(1 \pm \eta)$ and additive $O(w\eta)$ error, where $\eta = \tilde{O}\big(\sqrt{\ln(1/\delta)}/\varepsilon\big)$.

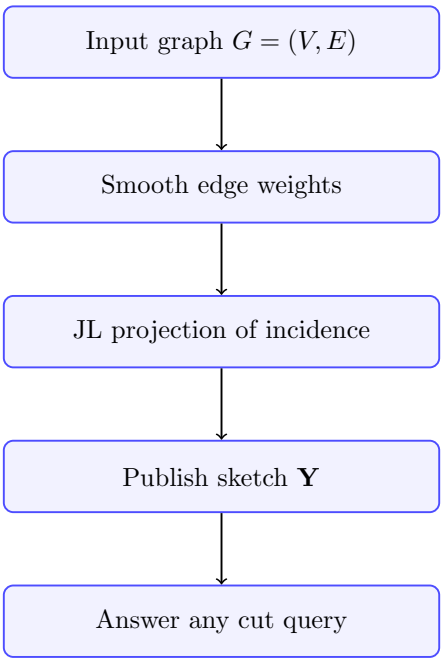


Figure 4: One-time JL sketch enables unlimited private cut queries.

Relationship to PrivateProjection.

- *Common core.* Both algorithms employ the *same* random-Gaussian projection (Lines 1 and 3).
- *Noise addition vs. constraint.* Kenthapadi *et al.* add Gaussian noise (Line 4); Blocki *et al.* omit this step but restrict inputs to rank-1 neighbouring changes and smooth the data to bound sensitivity.

- *Fallback rule.* If the spectral condition in Blocki's theorem fails, PrivateProjection is the safer choice; otherwise Blocki's sketch enjoys dimension-independent noise.

3.3 Fast JL Implementations

Table 1: Run-time and storage of common JL matrices (projecting an $n \times d$ matrix to $k = O(\log n/\varepsilon^2)$).

Variant	Time	Storage	Reference
Dense Gaussian	$O(ndk)$	$O(dk)$	classical JL
Sparse Sign (1/3 density)	$O(ndk)$	$O(dk)$	[1]
FastJL (FFT)	$O(nd \log k)$	$O(d)$	[2]
SRHT (Hadamard)	$O(nd \log d)$	$O(d)$	[8]

Replacing the dense Gaussian matrix in either Algorithm 1 or 2 by one of the above saves time and memory; privacy and utility guarantees remain up to constant factors (see Section 5).

4 Comparative Analysis

This section benchmarks JL-based methods against classic differential-privacy mechanisms along five axes: (i) core idea and data assumptions, (ii) error for graph cuts, (iii) error in distance estimation, (iv) query-interaction model, and (v) computational complexity.

4.1 Core Idea & Data Assumptions

Table 2: Mechanisms and their data assumptions.

Method	**Core Idea**	**Data Assumptions**
JL + Noise (Kenthapadi)	Random JL projection; add Gaussian noise calibrated to max row-norm.	Bounded ℓ_2-norm vectors; no preprocessing needed.
JL-Only (Blocki)	Single JL projection; inherent randomness suffices—no explicit noise.	Well-conditioned data (large singular values); rank-1 neighbouring changes.
Randomised Response	Flip each bit/coordinate with fixed probability.	Binary or bounded data; privacy cost scales with dimension.
Laplace / Gaussian	Compute query; add noise scaled to global sensitivity.	No structural assumptions, but sensitivity often $\Theta(d)$.
Multiplicative Weights (MWEM)	Iteratively fit synthetic data to noisy answers.	Fixed workload known in advance; works best for counting queries.

Take-away. JL methods gain most when data are high-dimensional but reasonably well-behaved (bounded norm or well-conditioned); classical input/output noise mechanisms work universally but

may add dimension-dependent noise.

4.2 Error on Graph-Cut Queries

Table 3: Error guarantees for answering *all* graph cuts.

Method	**Error Bound**	**Privacy Cost**
Laplace (per cut)	Add $\mathrm{Lap}(1/\varepsilon)$ on each edge; worst-case error $O\big(\lvert E\rvert/\varepsilon\big)$.	$(\varepsilon, 0)$-DP *per query*.
JL-Only (Blocki)	One sketch answers every cut with $O\big(\lvert S\rvert\eta\big)$ additive error, $\eta = \tilde{O}\big(\sqrt{\ln(1/\delta)}/\varepsilon\big)$.	Single (ε, δ)-DP budget for *unlimited* queries.

Observation. The one-time JL sketch is far better when many cut queries will be issued; Laplace dominates for a single query on a tiny graph.

4.3 Error in Distance Estimation

Table 4: Error when releasing pairwise squared distances.

Method	**Error Bound**	**Key Limitation**
Direct Noise	Gaussian noise on each squared distance: variance $O(d\sigma^2)$.	Error grows linearly with d.
Randomised Response	Bit-flip induces large bias for moderate distances.	Only binary (or heavily discretised) data.
JL + Noise (Kenthapadi)	After bias correction, MSE $O(k\sigma^2)$ with $k \ll d$.	Small residual multiplicative distortion.

4.4 Query-Interaction Model

Table 5: Interactivity, composition, and typical use-cases.

Method	Interaction Model	Composition	Typical Applications
Input Perturbation	Non-interactive release.	One-shot privacy loss.	Raw data sharing in federated analytics.
Output Perturbation	Interactive queries.	Budget consumed per query.	Exploratory data analysis.
MWEM	Interactive, iterative.	Advanced composition.	Large fixed workload of marginals.
JL + Noise	Non-interactive release.	Single budget.	Distance-based ML (clustering, k-NN).
JL-Only	Non-interactive release.	Single budget.	Graph analytics, PCA, covariance.

4.5 Computational Complexity

Table 6: Asymptotic complexity (dataset size $n \times d$).

Method	Time Complexity	Space Complexity
Laplace (full data)	$O(nd)$	$O(nd)$
MWEM (per iter.)	$O(n^2 d)$	$O(nd)$
JL + Noise (dense)	$O(ndk)$	$O(nk)$
JL + Noise (sparse / FastJL)	$O(ns \log d)$	$O(nk)$

Summary of Findings.

- **Utility vs. dimensionality.** JL reduces error from $O(d)$ to $O(k)$, a win when $k \ll d$.
- **Query volume.** Projection-based sketches shine when the analyst plans many downstream queries; per-query mechanisms lose cumulative budget.
- **Compute trade-offs.** Dense JL costs $O(ndk)$ time, but sparse FastJL lowers this to $O(ns \log d)$ while preserving privacy.
- **Assumption sensitivity.** Projection-only privacy relies on spectral conditions; when violated, projection + noise or classical mechanisms are safer.

5 Theoretical Guarantees

In this section we state the main privacy and utility theorems proved in our two JL-based mechanisms.

5.1 Privacy Guarantees

1. JL + Noise Mechanism (Kenthapadi et al. [7]) Let $x \in \mathbb{R}^d$ be any data vector with $\|x\|_2 \le 1$. Draw $P \in \mathbb{R}^{k\times d}$ with i.i.d. entries $P_{ij} \sim N(0, 1/k)$. Define the sketch

$$\tilde{y} \;=\; P\,x \;+\; \eta, \quad \eta \;\sim\; \mathcal{N}(0, \sigma^2 I_k).$$

Then for any target (ε, δ), if

$$\sigma \;\ge\; w_2(P)\,\frac{\sqrt{2\,\ln(1.25/\delta)}}{\varepsilon}, \quad \text{where} \quad w_2(P) = \max_{1\le i\le k} \|P_{(i)}\|_2,$$

the mapping $x \mapsto \tilde{y}$ satisfies (ε, δ)-differential privacy. Standard JL constructions ensure $w_2(P) \approx 1$, so in practice one sets $\sigma = O\big(\sqrt{\ln(1/\delta)}/\varepsilon\big)$ [7, Prop. 1].

2. JL-Only Mechanism (Blocki et al. [3]) Consider a data matrix $X \in \mathbb{R}^{n\times d}$ whose singular values are all at least $\gamma > 0$. Let $P \in \mathbb{R}^{k\times d}$ be drawn entrywise from $N(0, 1)$. Define the published matrix $Y = X\,P^\top$. Then if

$$k \;\ge\; C\,\frac{\ln(1/\delta)}{\varepsilon^2} \quad \text{and} \quad \gamma \;\ge\; \sqrt{\tfrac{2\,\ln(2/\delta)}{\varepsilon^2}},$$

for an absolute constant C, the single publication Y is (ε, δ)-differentially private for *all* linear queries on X (viewing neighboring datasets as differing by a rank-1 update) [3, Thm. 2.3]. No additional noise is required—the randomness of P alone suffices.

5.2 Utility Guarantees

3. Distance Preservation (JL + Noise) By the Johnson–Lindenstrauss lemma, with probability at least $1 - \beta$ a random $k \times d$ Gaussian projection preserves all pairwise distances among n points up to $(1 \pm \varepsilon)$ if

$$k \;\ge\; O\Big(\tfrac{\ln(n/\beta)}{\varepsilon^2}\Big).$$

After adding Gaussian noise of variance σ^2, the squared-distance estimator

$$\hat{D}_{ij} = \|\tilde{y}_i - \tilde{y}_j\|^2 - 2k\,\sigma^2$$

is unbiased: $\mathbb{E}[\hat{D}_{ij}] = \|x_i - x_j\|^2$, and concentrates tightly around its mean with variance $O(k\,\sigma^4 + \varepsilon^2\|x_i - x_j\|^4)$ [7, Lem. 3.2].

4. Linear Query Accuracy (JL-Only) Blocki et al. show that for any unit-norm linear functional $q \in \mathbb{R}^d$, the projected answer $q^\top Y$ differs from the true $q^\top X$ by at most a $(1 \pm \varepsilon)$ factor:

$$(1 - \varepsilon)\,\|X^\top q\|_2 \;\le\; \|Y^\top q\|_2 \;\le\; (1 + \varepsilon)\,\|X^\top q\|_2,$$

with high probability so long as $k = O(\ln(1/\delta)/\varepsilon^2)$ [3, Thm. 3.1]. This guarantees accurate answers for any family of linear queries—such as graph cut sizes or covariance entries—independent of n and d.

6 Open Problems & Future Work

Despite rapid progress, several questions remain open.

- **Beyond well-conditioned data.** Projection-only privacy (Blocki et al.) relies on a lower bound on singular values. Designing *spectral boosters* or adaptive pre-conditioning that preserve both utility and privacy is an active line of research.
- **Near-neighbour accuracy.** Current utility bounds degrade for very small true distances. Can variance-reduction or debiasing techniques tighten the additive error for closest-pair and clustering tasks?
- **Automatic parameter tuning.** Choosing k and noise scale σ is still heuristic in practice. A data-dependent but differentially-private procedure—for example via the *propose-test-release* paradigm—would be valuable.
- **Alternative metrics.** The JL lemma is Euclidean. Extending inherent privacy to cosine similarity, ℓ_1 distance, or earth-mover distance could unlock new domains such as NLP embeddings or computer-vision features.
- **Streaming and federated settings.** FastJL matrices already stream in $O(1)$ words per update; marrying them with continual-release DP (e.g. binary trees or sliding windows) remains largely unexplored.
- **Hardware-friendly projections.** Sparse integer or ternary JL matrices reduce multiplication cost; quantifying their privacy parameters under rounding error is an open engineering problem.

7 Conclusion

High-dimensional data analysis faces a dual challenge: the statistical *curse of dimensionality* and the legal *imperative of privacy*. This report surveyed how the Johnson–Lindenstrauss transform addresses both.

- We reviewed the classic JL lemma and the differential-privacy toolbox, establishing notation and sensitivity facts.
- We presented two projection-based mechanisms: PrivateProjection (projection + Gaussian noise) and JL-CutSketch (projection-only with smoothing). Formal theorems show each satisfies (ε, δ)-DP while preserving geometry up to controllable error.
- A comparative study demonstrated where JL methods out-perform input/output perturbation and multiplicative-weights schemes— especially in non-interactive, many-query, high-dimensional regimes.
- Fast, sparse, and Hadamard-based projections were catalogued, showing that privacy carries over with minor parameter tweaks.

Caveats. Projection-only privacy needs well-conditioned data; projection + noise incurs additive error that can hinder very fine-grained tasks. Moreover, JL currently caters to Euclidean geometry; other metrics remain open.

Why it matters. As organisations grapple with terabyte-scale feature spaces and stricter regulations (GDPR, CCPA), JL-based differentially-private sketches offer a rare combination of *simplicity, scalability,* and *theoretical guarantees*. Continued progress on the open problems above will determine whether these techniques become the default privacy layer for high-dimensional machine-learning pipelines.

Project Links

Overleaf Project Report:
https://www.overleaf.com/3936579644dhpqsgztvzdb#c69b53

Overleaf Presentation Slides:
https://www.overleaf.com/3257915794qyyqvrxqkhpq#5bcda1

References

[1] Dimitris Achlioptas. Database-friendly random projections: Johnson–lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[2] Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson–lindenstrauss transform. In *Proc. 38th ACM Symposium on Theory of Computing (STOC)*, pages 557–563, 2006.

[3] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The Johnson–Lindenstrauss transform itself preserves differential privacy. In *53rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 410–419. IEEE, 2012.

[4] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. 3rd Theory of Cryptography Conference (TCC)*, pages 265–284. Springer, 2006.

[5] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*, volume 9 of *Foundations and Trends in Theoretical Computer Science*. Now Publishers, 2014.

[6] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[7] Kamesh Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. Privacy via the Johnson–Lindenstrauss transform. *Journal of Privacy and Confidentiality*, 5(1):39–71, 2013.

[8] Joel A. Tropp. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(1–2):115–126, 2011.