

Lectures 10 & 11: Vector and Matrix Quantization in LLMs

Instructors: *Alex Andoni, Edo Liberty*Lecturer: *Edo Liberty*

Lecture Overview & Pedagogical Roadmap

Modern Large Language Models (LLMs) and retrieval systems store and stream massive collections of high-dimensional vectors: model weight matrices, token embeddings, and key-value (KV) caches. In previous lectures, we studied how to reduce the **number of vectors** via Approximate Nearest Neighbor (ANN) search and coresets, and how to reduce the **number of floats per vector** via Johnson-Lindenstrauss random projections and low-rank Singular Value Decomposition (SVD).

In these two 90-minute sessions, we study the third fundamental axis of compression—**Quantization**: reducing the **number of bits per float**.

1. **Session 1 — Foundations, Scalar Quantization & 2D Gains (Sections 1–4):** We introduce the language of lossy compression and **Rate-Distortion Theory** (codebooks, codewords, decision regions, rate R , and MSE distortion D), and work out exact 1D quantizers for uniform and Gaussian distributions (**Section 2**). We then show how pre-multiplying an arbitrary vector $x \in \mathbb{R}^d$ by a random rotation makes all of its coordinates identically distributed and close to Gaussian, unlocking universal 1D scalar quantization as used in **EDEN** [10], **RaBitQ** [1], and **TurboQuant** [12] (**Section 3**). Finally, by moving from 1D to 2D and comparing square vs. hexagonal tilings of the plane, we prove from scratch that **no scalar quantizer can ever be optimal**, even for independent coordinates (**Section 4**).
2. **Session 2 — Vector Quantization Limits, k -Means, Product Quantization & SVD (Sections 5–8):** We prove the fundamental $R(D) = \frac{1}{2} \log_2(\sigma^2/D)$ **Rate-Distortion limit for high-dimensional Gaussians** via a geometric sphere-covering volume argument (**Section 5**), and prove the optimality conditions of **k -means (Lloyd-Max)** [6, 8] for any distribution (**Section 6**). Because a rate- R codebook in $\mathbb{R}^d$ requires $K = 2^{dR}$ centroids—making direct k -means impossible in high dimensions—we introduce **Product Quantization (PQ)** [5] (**Section 7**). Finally, using **Zador’s scaling** [11, 3] ($D_{VQ} \propto (\prod_{i=1}^d \sigma_i)^{2/d}$), we prove deductively via **concavity over the majorization polytope** that the optimal pre-rotation for Product Quantization is $R = PU^T$ —showing it is always optimal to **SVD the matrix first** and assign coordinates into groups of size d/m (**Section 8**).

Session 1: Foundations, Scalar Quantization & 2D Gains

1 Motivation: The Three Axes of Compression in LLMs

Consider any collection of N vectors $x_1, \dots, x_N \in \mathbb{R}^d$ in a modern AI system—such as the rows of a model weight matrix, a corpus of document embeddings in RAG, or the cached key and value vectors $(k_t, v_t)_{t=1}^N$ in an LLM’s KV-cache. If each floating-point coordinate is represented using b bits (e.g., $b = 16$ for FP16/BF16 or $b = 32$ for FP32), the total memory footprint is:

$$\text{Total Memory} = \underbrace{N}_{\# \text{ vectors}} \times \underbrace{d}_{\# \text{ floats / vector}} \times \underbrace{b}_{\# \text{ bits / float}} \text{ bits.} \quad (1)$$

Why This Bottlenecks Modern LLMs. In a frontier model such as **Llama-3 70B** (7×10^{10} parameters), storing the weights in FP16 ($b = 16$) requires 140 GB of GPU High-Bandwidth Memory (HBM)—exceeding a single 80 GB H100 GPU. Simultaneously, at a context length of $N = 128\text{k}$ tokens, the KV-cache alone consumes 43 GB per request. During autoregressive token generation, every single generated token requires streaming all model weights and all cached KV vectors from HBM to on-chip SRAM, making inference strictly **memory-bandwidth bound**.

To compress $N \times d \times b$, our course attacks each of the three factors in (1) sequentially:

Compression Axis	Target Factor	Algorithmic Toolkit	Lectures
1. Vector Search & Pruning	Reduce # of vectors $N \rightarrow k \ll N$	Locality-Sensitive Hashing (LSH), ANN indices, Coresets, Heavy-Hitter Eviction	Lectures 4–7
2. Dimension Reduction	Reduce # of floats per vector $d \rightarrow k \ll d$	Johnson-Lindenstrauss (JL) Random Projections, Randomized SVD ($A_k = U_k \Sigma_k V_k^T$)	Lectures 8–9
3. Quantization	Reduce # of bits per float $b \rightarrow R \ll 16$	Scalar Quantization, Random Rotations, Vector Quantization (k -Means), Product Quantization	Lectures 10–11

Notice that these three axes are **orthogonal and composable**: after selecting the relevant vectors and projecting them to the right subspace, we still need to represent each real number using as few bits R as possible (e.g., $R = 4, 2, 1$, or even $R < 1$ bits per coordinate).

2 Rate-Distortion Basics: Concepts, Terminology & 1D Examples

2.1 Why Lossy Compression? Terminology & Formal Definitions

A continuous random variable $X \in \mathbb{R}$ takes uncountably many values and therefore contains an **infinite** amount of information: specifying X exactly requires an infinite binary expansion. To represent X using a finite number of bits, we must map continuous values to a finite set of discrete reconstruction values—an inherently **lossy** operation called **quantization**.

Continuous Source $X \in \mathbb{R} \implies [\textbf{Quantizer} / \textbf{Encoder}] \implies \text{Index } i \in \{1, \dots, K\} \text{ (} R = \log_2 K \text{ bits)}$ $\text{Index } i \text{ (} R \text{ bits)} \implies [\textbf{Codebook} / \textbf{Decoder}] \implies \text{Reconstruction } \hat{X} = y_i \in \mathcal{C}$

Definition 2.1 (Scalar Quantizer, Codebook, Rate, and Distortion). *Let $X \in \mathbb{R}$ be a real-valued random variable with probability density function $p(x)$. A K -level scalar quantizer $Q : \mathbb{R} \rightarrow \mathcal{C}$ consists of:*

1. **Codebook (or Dictionary) $\mathcal{C}$:** *A finite set of K reconstruction values $\mathcal{C} = \{y_1, y_2, \dots, y_K\} \subset \mathbb{R}$, where each y_i is called a **codeword** (or **centroid**).*
2. **Decision Regions (Partition) $\{S_1, \dots, S_K\}$:** *A partition of the real line into K disjoint intervals $\bigcup_{i=1}^K S_i = \mathbb{R}$ ($S_i \cap S_j = \emptyset$ for $i \neq j$), such that the quantizer maps every input $x \in S_i$ to its region's codeword:*

$$Q(x) = y_i \quad \text{whenever } x \in S_i.$$

3. **Rate (R):** *Using fixed-length binary indices to specify which of the K codewords was chosen requires:*

$$R \triangleq \log_2 K \quad \text{bits per symbol} \quad \iff \quad K = 2^R. \quad (2)$$

4. **Distortion (D):** *The expected error between the source X and its quantized reconstruction $\hat{X} = Q(X)$. Throughout these notes, we measure distortion by the **Mean Squared Error (MSE)**:*

$$D \triangleq \mathbb{E}[(X - Q(X))^2] = \sum_{i=1}^K \int_{S_i} (x - y_i)^2 p(x) dx. \quad (3)$$

The Rate-Distortion Function. For a given source distribution $p(x)$, the fundamental question of lossy compression is: *What is the minimum possible distortion $D(R)$ achievable at rate R bits per symbol (or equivalently, the minimum rate $R(D)$ required to achieve distortion at most D)?*

2.2 Worked Example 1: Uniform Distribution $X \sim \text{Unif}[-a, a]$

Suppose X is uniformly distributed on the interval $[-a, a]$, so its density is $p(x) = \frac{1}{2a}$ for $x \in [-a, a]$ (and 0 otherwise), with variance $\mathbb{E}[X^2] = \frac{a^2}{3}$.

Given a rate of R bits ($K = 2^R$ codewords), symmetry and uniformity of $p(x)$ suggest partitioning $[-a, a]$ into $K = 2^R$ equal sub-intervals $S_1, \dots, S_K$, each of width (step size):

$$\Delta = \frac{2a}{K} = \frac{2a}{2^R}, \quad (4)$$

and placing each codeword y_i at the **midpoint** of its interval $S_i = [y_i - \frac{\Delta}{2}, y_i + \frac{\Delta}{2}]$.

Claim 2.2 (Rate-Distortion of Uniform Scalar Quantization). *For $X \sim \text{Unif}[-a, a]$, the $K = 2^R$ -level uniform midpoint quantizer with step size $\Delta = 2a/2^R$ achieves Mean Squared Error:*

$$D(R) = \frac{\Delta^2}{12} = \frac{a^2}{3} 2^{-2R} = \text{Var}(X) \cdot 4^{-R}, \quad \text{or equivalently} \quad R(D) = \frac{1}{2} \log_2 \left(\frac{\text{Var}(X)}{D} \right). \quad (5)$$

Proof. Within each interval $S_i = [y_i - \frac{\Delta}{2}, y_i + \frac{\Delta}{2}]$, change variables to the local quantization error $z = x - y_i \in [-\frac{\Delta}{2}, \frac{\Delta}{2}]$. Summing over all K intervals of length Δ with $K\Delta = 2a$:

$$\begin{aligned} D(R) &= \sum_{i=1}^K \int_{y_i - \Delta/2}^{y_i + \Delta/2} (x - y_i)^2 \frac{1}{2a} dx = \frac{K}{2a} \int_{-\Delta/2}^{\Delta/2} z^2 dz \\ &= \frac{1}{\Delta} \left[\frac{z^3}{3} \right]_{-\Delta/2}^{\Delta/2} = \frac{1}{\Delta} \left(\frac{\Delta^3}{24} - \left(-\frac{\Delta^3}{24} \right) \right) = \frac{\Delta^2}{12}. \end{aligned}$$

Substituting $\Delta = 2a \cdot 2^{-R}$ gives $D(R) = \frac{4a^2}{12} 2^{-2R} = \frac{a^2}{3} 2^{-2R}$. Solving for R yields $2^{2R} = \frac{a^2/3}{D} \implies R(D) = \frac{1}{2} \log_2 \left(\frac{a^2/3}{D} \right)$. $\square$

Key Takeaway. Every additional bit of rate ($R \leftarrow R + 1$) halves the bin width Δ and reduces the Mean Squared Error $D(R)$ by a **factor of 4** (≈ 6 dB per bit).

2.3 Worked Example 2: Gaussian Distribution $X \sim \mathcal{N}(0, 1)$ at $R = 1$ Bit

Now suppose $X \sim \mathcal{N}(0, 1)$ is a standard Gaussian with density $p(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$, and suppose we are given a budget of just $R = 1$ **bit** ($K = 2^1 = 2$ codewords).

1. **Choosing the Decision Threshold:** Because $p(x)$ is symmetric around 0, the two decision regions must split at 0: $S_1 = (-\infty, 0)$ and $S_2 = [0, \infty)$. Thus the encoder simply transmits the **sign** of X , and the codebook is symmetric: $\mathcal{C} = \{-c, +c\}$ for some reconstruction level $c > 0$, i.e., $Q(x) = c \cdot \text{sign}(x)$.
2. **Finding the Optimal Reconstruction Codeword c^* :** What value of c minimizes the Mean Squared Error $D(c) = \mathbb{E}[(X - Q(X))^2]$?

Claim 2.3 (Optimal 1-Bit Scalar Quantizer for a Gaussian). *For $X \sim \mathcal{N}(0, 1)$ and decision regions $S_1 = (-\infty, 0)$, $S_2 = [0, \infty)$, the optimal reconstruction codebook is:*

$$\mathcal{C}^* = \left\{ -\sqrt{\frac{2}{\pi}}, +\sqrt{\frac{2}{\pi}} \right\} \approx \{-0.7979, +0.7979\}, \quad (6)$$

achieving Mean Squared Error:

$$D_{SQ}(1) = 1 - \frac{2}{\pi} \approx 0.3634. \quad (7)$$

More generally, for $X \sim \mathcal{N}(0, \sigma^2)$, scaling by σ gives $\mathcal{C}^* = \{\pm\sigma\sqrt{2/\pi}\}$ and $D_{SQ}(1) = (1 - \frac{2}{\pi})\sigma^2$.

Proof. Step 1 (Symmetry & Conditional Expectation): By symmetry of the Gaussian density around 0, the MSE on $(-\infty, 0)$ equals the MSE on $[0, \infty)$:

$$D(c) = \int_{-\infty}^0 (x + c)^2 p(x) dx + \int_0^{\infty} (x - c)^2 p(x) dx = 2 \int_0^{\infty} (x - c)^2 p(x) dx = \mathbb{E}[(X - c)^2 \mid X \geq 0].$$

Differentiating with respect to c and setting $\frac{d}{dc} D(c) = 0$ gives the Minimum Mean Squared Error (MMSE) condition:

$$c^* = \mathbb{E}[X \mid X \geq 0] = \frac{\int_0^{\infty} x p(x) dx}{\Pr[X \geq 0]}.$$

Step 2 (Evaluating the Gaussian Integral): Since $\Pr[X \geq 0] = \frac{1}{2}$ and $p(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$:

$$c^* = 2 \int_0^\infty x \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = \sqrt{\frac{2}{\pi}} \int_0^\infty x e^{-x^2/2} dx = \sqrt{\frac{2}{\pi}} \left[-e^{-x^2/2} \right]_0^\infty = \sqrt{\frac{2}{\pi}} (0 - (-1)) = \sqrt{\frac{2}{\pi}}.$$

Step 3 (Evaluating the Resulting Distortion): Expanding $(X - c^*)^2 = X^2 - 2c^*X + (c^*)^2$ conditioned on $X \geq 0$ (where $\mathbb{E}[X^2 \mid X \geq 0] = 1$ and $\mathbb{E}[X \mid X \geq 0] = c^*$):

$$D(c^*) = \mathbb{E}[X^2 \mid X \geq 0] - 2c^*\mathbb{E}[X \mid X \geq 0] + (c^*)^2 = 1 - (c^*)^2 = 1 - \frac{2}{\pi} \approx 0.3634. \quad \square$$

How Optimal Is 0.3634? In Session 2, we will prove that the theoretical **Shannon Rate-Distortion limit** for $\mathcal{N}(0, 1)$ at rate $R = 1$ bit is $D^*(1) = 2^{-2(1)} = 0.25$. Notice that 1D scalar quantization (0.3634) is nearly 50% above the fundamental limit (0.25).

3 Random Rotation + Scalar Quantization in High Dimensions

How do we apply scalar quantization to an **arbitrary** vector $x \in \mathbb{R}^d$ (such as a KV-cache key/value vector or a layer weight row) when we do not know the distribution of its coordinates—or worse, when x contains massive **outlier coordinates**?

Why Naive Coordinate-Wise Quantization Fails on Outliers. Consider two unit vectors in $\mathbb{R}^d$ ($\|x\|_2 = 1$):

1. A *flat* vector $x_{\text{flat}} = (\pm \frac{1}{\sqrt{d}}, \dots, \pm \frac{1}{\sqrt{d}})^T$, whose coordinates all lie in $[-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}]$.
2. A *1-sparse outlier* vector $x_{\text{outlier}} = (1, 0, \dots, 0)^T$.

If we apply a uniform grid on $[-\|x\|_\infty, \|x\|_\infty]$ to x_{outlier} , the grid must span $[-1, 1]$ just to cover the single outlier coordinate $x_1 = 1$, while the other $d - 1$ coordinates sit at 0 and waste all remaining grid levels—inflating the total squared error $\|x - Q(x)\|_2^2$ by a factor of $\Theta(d)$.

3.1 Preconditioning via Random Rotation

To eliminate outliers and unknown coordinate distributions, we apply a **random orthogonal rotation** $U \in O(d)$ ($U^T U = U U^T = I_d$) before quantizing. Because U is orthogonal, it preserves all Euclidean norms, distances, and inner products exactly:

$$\|Ux\|_2 = \|x\|_2, \quad \|Ux - Uy\|_2 = \|x - y\|_2, \quad \langle Ux, Uy \rangle = \langle x, y \rangle. \quad (8)$$

Claim 3.1 (Random Rotation Makes Coordinates Identically Distributed and Nearly Gaussian — Stated Without Proof). *Let $x \in \mathbb{R}^d$ be any fixed vector and let $U \in O(d)$ be drawn uniformly at random from the orthogonal group. Then the rotated vector $\tilde{x} = Ux$ is distributed uniformly on the sphere of radius $\|x\|_2$, and therefore:*

1. **Identical Distribution:** All d coordinates $\tilde{x}_1, \dots, \tilde{x}_d$ have the **exact same marginal distribution** with mean $\mathbb{E}[\tilde{x}_j] = 0$ and variance $\mathbb{E}[\tilde{x}_j^2] = \frac{\|x\|_2^2}{d}$.

2. **Asymptotic Gaussianity:** As the dimension d grows, the marginal distribution of every coordinate $\tilde{x}_j$ converges to a Gaussian distribution:

$$\tilde{x}_j = (Ux)_j \stackrel{d}{\approx} \mathcal{N}\left(0, \frac{\|x\|_2^2}{d}\right) \quad \forall j \in \{1, \dots, d\}. \quad (9)$$

(Intuition from Lecture 8: Each coordinate $\tilde{x}_j = \langle u_j, x \rangle$ is the inner product of x with a random unit vector $u_j \approx \frac{1}{\sqrt{d}}g$ where $g \sim \mathcal{N}(0, I_d)$; by rotational invariance of Gaussians, $\langle g, x \rangle \sim \mathcal{N}(0, \|x\|_2^2)$, so $\tilde{x}_j \approx \mathcal{N}(0, \|x\|_2^2/d)$ regardless of how spiky or adversarial x was.)

3.2 The Rotate-and-Scalar-Quantize Paradigm in Modern AI

Claim 3.1 reduces the quantization of **any** worst-case vector $x \in \mathbb{R}^d$ to the 1D Gaussian scalar quantization problem we solved in Section 2:

Algorithm ROTATED-SCALAR-QUANTIZE($x \in \mathbb{R}^d$, shared random rotation $U \in O(d)$, rate R)

1. Record norm $r \leftarrow \|x\|_2$ and normalize $u \leftarrow x/r$.
2. Rotate: $\tilde{u} \leftarrow Uu$ (so each coordinate $\tilde{u}_j \stackrel{d}{\approx} \mathcal{N}(0, 1/d)$).
3. Quantize each coordinate independently using the optimal R -bit 1D $\mathcal{N}(0, 1/d)$ scalar quantizer Q_R :
$$z_j \leftarrow Q_R(\tilde{u}_j) \quad \text{for } j = 1, \dots, d.$$
4. **To reconstruct:** return $\hat{x} = r \cdot U^T z$ (or compute inner products $\langle q, \hat{x} \rangle = r \cdot \langle Uq, z \rangle$ directly in rotated space).

By linearity of expectation and orthogonality of U , the total expected squared Euclidean error for **any** vector $x \in \mathbb{R}^d$ at $R = 1$ bit is simply the sum of the d coordinate-wise Gaussian distortions from Claim 2.3:

$$\mathbb{E}[\|x - \hat{x}\|_2^2] = \mathbb{E}[\|Ux - rz\|_2^2] = \sum_{j=1}^d \mathbb{E}[(\tilde{x}_j - rz_j)^2] \approx \sum_{j=1}^d \left(1 - \frac{2}{\pi}\right) \frac{\|x\|_2^2}{d} = \left(1 - \frac{2}{\pi}\right) \|x\|_2^2 \approx 0.3634 \|x\|_2^2.$$

In practice, instead of a dense $\mathcal{O}(d^2)$ matrix U , systems use a **Randomized Walsh-Hadamard Transform** $U = \frac{1}{\sqrt{d}}HD$ with random ± 1 signs D , which runs in $\mathcal{O}(d \log d)$ time and $\mathcal{O}(d)$ memory.

Key Literature Citations Using This Paradigm.

1. **EDEN [10]:** Rotates client gradient vectors with a randomized Hadamard transform so coordinates become approximately i.i.d. Gaussian, then applies scalar quantization and entropy coding for communication-efficient distributed mean estimation.

2. **RaBitQ [1]**: Applies a random orthogonal rotation followed by 1-bit sign quantization (and multi-bit extensions) with an unbiased inner-product estimator for high-dimensional Approximate Nearest Neighbor search.
3. **TurboQuant [12]**: Applies a random rotation to LLM KV-cache and weight vectors followed by pre-computed optimal Gaussian codebooks, achieving near-optimal rate-distortion online without per-model calibration.

4 Showing Gains in 2D: Proving That No Scalar Quantizer Can Be Optimal

So far, we have quantized each coordinate $X_1, \dots, X_d$ **independently** (Scalar Quantization). Can we do strictly better by quantizing multiple coordinates **jointly** as a vector?

4.1 Formalizing Vector Quantization in k Dimensions (Focusing on $k = 2$)

Suppose we group coordinates into pairs $X = (X_1, X_2) \in \mathbb{R}^2$ (or more generally, blocks of k coordinates in $\mathbb{R}^k$).

Definition 4.1 (Vector Quantizer & Per-Coordinate Rate). *A k -dimensional Vector Quantizer (VQ) $Q : \mathbb{R}^k \rightarrow \mathcal{C}$ maps each vector $x \in \mathbb{R}^k$ to one of K codewords in a codebook $\mathcal{C} = \{y_1, \dots, y_K\} \subset \mathbb{R}^k$, partitioning $\mathbb{R}^k$ into K disjoint decision regions $S_1, \dots, S_K \subset \mathbb{R}^k$. Because $\log_2 K$ bits represent k coordinates simultaneously, the **rate per coordinate** is:*

$$R = \frac{\log_2 K}{k} \quad \text{bits/coordinate} \quad \Longleftrightarrow \quad K = 2^{kR}, \quad (10)$$

and the **per-coordinate MSE distortion** is $D = \frac{1}{k} \mathbb{E}[\|X - Q(X)\|_2^2]$.

In $k = 2$ dimensions at rate R bits/coordinate, both a scalar quantizer and a 2D vector quantizer are allowed $K = 2^{2R} = (2^R)^2$ codewords in $\mathbb{R}^2$. However:

- A **Scalar Quantizer** $Q(x_1, x_2) = (Q_1(x_1), Q_2(x_2))$ is restricted to a Cartesian grid $\mathcal{C}_1 \times \mathcal{C}_2$ whose decision regions $S_{ij} = I_i \times J_j$ must be **axis-aligned rectangles**.
- A **2D Vector Quantizer** can place its $K = 2^{2R}$ codewords anywhere in $\mathbb{R}^2$ and partition the plane into **arbitrary 2D decision regions** (e.g., hexagons or rotated cells).

4.2 Gain 1: Exploiting Dependence Between Coordinates

Consider a 2D distribution where (X_1, X_2) is uniformly distributed on the two quadrants where X_1 and X_2 have the **same sign**: $[-a, 0]^2 \cup [0, a]^2$, and has zero probability in the other two quadrants ($\text{sign}(X_1) \neq \text{sign}(X_2)$).

- **Scalar Quantization (6×6 Grid, $K_{\text{SQ}} = 36$)**: If we quantize X_1 and X_2 independently into 6 uniform intervals each, the Cartesian product creates $6 \times 6 = 36$ square decision regions of side length $\Delta = a/3$, requiring rate:

$$R_{\text{SQ}} = \frac{\log_2(36)}{2} = \log_2 6 \approx 2.585 \text{ bits/coordinate.}$$

However, 18 of those 36 squares lie in the second and fourth quadrants where (X_1, X_2) **never occurs**.

- **2D Vector Quantization** ($K_{\text{VQ}} = 18$): A 2D vector quantizer simply keeps the 18 squares in the first and third quadrants and discards the 18 empty squares. It achieves the **exact same MSE distortion** ($\Delta^2/12$ per coordinate) using only half the codewords ($K_{\text{VQ}} = 18$), reducing the rate to:

$$R_{\text{VQ}} = \frac{\log_2(18)}{2} = \frac{\log_2(36/2)}{2} = \log_2 6 - \frac{1}{2} = R_{\text{SQ}} - 0.5 \text{ bits/coordinate.}$$

4.3 Gain 2: Even for Independent Coordinates! (The Square vs. Hexagon Proof)

What if X_1 and X_2 are **completely independent** (for example, i.i.d. uniform on a large region $[-a, a]^2$)? One might guess that when $X_1 \perp X_2$, quantizing X_1 and X_2 independently via scalar quantization is optimal. **Remarkably, this is false!**

Theorem 4.2 (Strict Suboptimality of Scalar Quantization in 2D). *Let (X_1, X_2) be uniformly distributed on $\mathbb{R}^2$. Comparing a scalar quantizer (which tiles $\mathbb{R}^2$ with **squares** of area A) against a 2D vector quantizer that tiles $\mathbb{R}^2$ with **regular hexagons** of the **exact same area** A (and hence the exact same rate R):*

$$\text{MSE}_{\text{square}} = \frac{A}{6} \approx 0.1667A, \quad \text{whereas} \quad \text{MSE}_{\text{hex}} = \frac{5}{18\sqrt{3}}A \approx 0.1604A. \quad (11)$$

Thus, the hexagonal 2D vector quantizer achieves strictly lower distortion at the same rate:

$$\frac{\text{MSE}_{\text{hex}}}{\text{MSE}_{\text{square}}} = \frac{5}{3\sqrt{3}} \approx 0.9622 \quad (\text{a } 3.8\% \text{ reduction in MSE}). \quad (12)$$

*Consequently, **no scalar quantizer can be optimal**, even when coordinates are independent.*

Proof. To compare the two quantizers at the **same rate** R , both must use the same number of codewords per unit area, which means each decision cell must have the **same 2D area** A . Within any cell $S \subset \mathbb{R}^2$ of area A centered at the origin $(0,0)$ under a uniform density $p(x, y) = 1/A$, the total 2D Mean Squared Error is:

$$\text{MSE}(S) = \frac{1}{A} \iint_S (x^2 + y^2) dx dy.$$

Step 1 (MSE of a Square Decision Region of Side Length Δ): A scalar quantizer with step size Δ has square decision regions $S_{\text{square}} = [-\frac{\Delta}{2}, \frac{\Delta}{2}]^2$ of area:

$$A = \Delta^2. \quad (13)$$

Integrating $x^2 + y^2$ over the square gives:

$$\begin{aligned}
\iint_{S_{\text{square}}} (x^2 + y^2) dx dy &= \int_{-\Delta/2}^{\Delta/2} \int_{-\Delta/2}^{\Delta/2} (x^2 + y^2) dx dy \\
&= \int_{-\Delta/2}^{\Delta/2} \left[\frac{x^3}{3} + xy^2 \right]_{x=-\Delta/2}^{x=\Delta/2} dy \\
&= \int_{-\Delta/2}^{\Delta/2} \left(\frac{\Delta^3}{12} + \Delta y^2 \right) dy \\
&= \left[\frac{\Delta^3}{12} y + \Delta \frac{y^3}{3} \right]_{y=-\Delta/2}^{y=\Delta/2} = \frac{\Delta^4}{12} + \frac{\Delta^4}{12} = \frac{\Delta^4}{6}.
\end{aligned} \tag{14}$$

Dividing by $A = \Delta^2$ yields the MSE of the square cell in terms of its area A :

$$\text{MSE}_{\text{square}} = \frac{1}{A} \cdot \frac{\Delta^4}{6} = \frac{\Delta^2}{6} = \frac{A}{6}. \tag{15}$$

Step 2 (Area and MSE of a Regular Hexagonal Decision Region of Side Length δ): Now tile $\mathbb{R}^2$ with regular hexagons S_{hex} of side length (circumradius) δ . By symmetry, a regular hexagon centered at the origin is the union of 6 congruent equilateral triangles of side length δ . Consider the top triangle T , which has height $h = \frac{\sqrt{3}}{2}\delta$ along the y -axis and base width δ at $y = \frac{\sqrt{3}}{2}\delta$:

$$T = \left\{ (x, y) \in \mathbb{R}^2 : 0 \leq y \leq \frac{\sqrt{3}}{2}\delta, \quad -\frac{y}{\sqrt{3}} \leq x \leq \frac{y}{\sqrt{3}} \right\}.$$

- *Area of the Hexagon:* Since each triangle T has area $\frac{1}{2} \cdot \text{base} \cdot \text{height} = \frac{1}{2} \cdot \delta \cdot \frac{\sqrt{3}}{2}\delta = \frac{\sqrt{3}}{4}\delta^2$, the area of the hexagon is:

$$A = 6 \cdot \frac{\sqrt{3}}{4}\delta^2 = \frac{3\sqrt{3}}{2}\delta^2. \tag{16}$$

- *Integral of $x^2 + y^2$ over the Hexagon:* By 6-fold symmetry, $\iint_{S_{\text{hex}}} (x^2 + y^2) dx dy = 6 \iint_T (x^2 + y^2) dx dy$. Evaluating the integral over T :

$$\begin{aligned}
\iint_T (x^2 + y^2) dx dy &= \int_0^{\frac{\sqrt{3}}{2}\delta} \left(\int_{-y/\sqrt{3}}^{y/\sqrt{3}} (x^2 + y^2) dx \right) dy \\
&= \int_0^{\frac{\sqrt{3}}{2}\delta} \left[\frac{x^3}{3} + xy^2 \right]_{x=-y/\sqrt{3}}^{x=y/\sqrt{3}} dy \\
&= \int_0^{\frac{\sqrt{3}}{2}\delta} \left(\frac{2y^3}{9\sqrt{3}} + \frac{2y^3}{\sqrt{3}} \right) dy = \frac{20}{9\sqrt{3}} \int_0^{\frac{\sqrt{3}}{2}\delta} y^3 dy \\
&= \frac{20}{9\sqrt{3}} \cdot \frac{1}{4} \left(\frac{\sqrt{3}}{2}\delta \right)^4 = \frac{20}{9\sqrt{3}} \cdot \frac{9}{64} \delta^4 = \frac{5\sqrt{3}}{48} \delta^4.
\end{aligned}$$

Multiplying by 6 triangles gives the integral over the full hexagon:

$$\iint_{S_{\text{hex}}} (x^2 + y^2) dx dy = 6 \cdot \frac{5\sqrt{3}}{48} \delta^4 = \frac{5\sqrt{3}}{8} \delta^4. \quad (17)$$

Step 3 (Comparing MSE_{hex} and $\text{MSE}_{\text{square}}$ at Equal Area A): Dividing (17) by the hexagon area $A = \frac{3\sqrt{3}}{2} \delta^2$ and substituting $\delta^2 = \frac{2A}{3\sqrt{3}}$ from (16):

$$\text{MSE}_{\text{hex}} = \frac{1}{A} \cdot \frac{5\sqrt{3}}{8} \delta^4 = \frac{\frac{5\sqrt{3}}{8} \delta^4}{\frac{3\sqrt{3}}{2} \delta^2} = \frac{5}{12} \delta^2 = \frac{5}{12} \cdot \frac{2A}{3\sqrt{3}} = \frac{5}{18\sqrt{3}} A.$$

Comparing with $\text{MSE}_{\text{square}} = \frac{A}{6}$ from (15) gives the exact ratio:

$$\frac{\text{MSE}_{\text{hex}}}{\text{MSE}_{\text{square}}} = \frac{\frac{5}{18\sqrt{3}} A}{\frac{1}{6} A} = \frac{5}{3\sqrt{3}} \approx 0.96225 < 1. \quad \square$$

Geometric Intuition (The Space-Filling Advantage of High Dimensions). Among all 2D shapes of fixed area A , the **circle** minimizes the second moment $\frac{1}{A} \iint (x^2 + y^2) dx dy = \frac{A}{2\pi} \approx 0.1592A$ because all boundary points are equidistant from the center, avoiding the “distant corners” of a square. While circles cannot tile the 2D plane without gaps, **regular hexagons** are much closer to circles than squares are. In higher dimensions $d \rightarrow \infty$, high-dimensional lattice polytopes become asymptotically spherical, approaching the $\frac{1}{2\pi e}$ sphere limit [11, 3].

Session 2: Rate-Distortion Limits, k -Means, Product Quantization & SVD

5 Rate-Distortion Limit Proof for Random Gaussians

How much further can we reduce distortion as the vector dimension d grows? We now establish the fundamental **Rate-Distortion limit** for a d -dimensional Gaussian source $X = (X_1, \dots, X_d)^T \sim \mathcal{N}(0, \sigma^2 I_d)$ using a geometric sphere-covering argument.

Theorem 5.1 (Rate-Distortion Lower Bound for Gaussian Vectors [9]). *Let $X = (X_1, \dots, X_d)^T \sim \mathcal{N}(0, \sigma^2 I_d)$ have i.i.d. Gaussian coordinates of variance σ^2 . For any vector quantizer $Q : \mathbb{R}^d \rightarrow \mathcal{C}$ with rate $R = \frac{1}{d} \log_2 |\mathcal{C}|$ bits per coordinate that achieves per-coordinate MSE distortion $D = \frac{1}{d} \mathbb{E}[\|X - Q(X)\|_2^2] \leq \sigma^2$ in the high-dimensional regime ($d \rightarrow \infty$), the rate R and distortion D must satisfy:*

$$R(D) \geq \frac{1}{2} \log_2 \left(\frac{\sigma^2}{D} \right) \quad \Longleftrightarrow \quad D(R) \leq \sigma^2 2^{-2R}. \quad (18)$$

(For $D > \sigma^2$, the single codeword $\mathcal{C} = \{0\}$ trivially achieves $D = \sigma^2$ with $R = 0$ bits, so $R(D) = \max\{0, \frac{1}{2} \log_2(\sigma^2/D)\}$.)

Proof. Step 1 (High-Dimensional Gaussian Vectors Concentrate on a Sphere of Radius $\sigma\sqrt{d}$): Because $X_1, \dots, X_d \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ with $\mathbb{E}[X_i^2] = \sigma^2$, the Law of Large Numbers (and the χ_d^2 concentration proved in Lecture 8) implies that as d grows, the squared Euclidean norm of X concentrates tightly around $d\sigma^2$:

$$\|X\|_2 = \sqrt{\sum_{i=1}^d X_i^2} \approx \sigma\sqrt{d} \quad \text{with high probability.} \quad (19)$$

Furthermore, the Gaussian density $p(x) = (2\pi\sigma^2)^{-d/2} e^{-\|x\|_2^2/(2\sigma^2)}$ is rotationally invariant and nearly uniform over the thin spherical shell of radius $r_{\text{source}} = \sigma\sqrt{d}$. Thus, any quantizer must cover the d -dimensional Euclidean ball $B_2^d(0, \sigma\sqrt{d})$ of radius $\sigma\sqrt{d}$.

Step 2 (Each Codeword Covers a Ball of Radius $\sqrt{dD}$): Let $\mathcal{C} = \{y_1, \dots, y_K\} \subset \mathbb{R}^d$ be the codebook of size $K = 2^{dR}$. For a source vector $x \in \mathbb{R}^d$ quantized to codeword y_i to incur per-coordinate squared error at most D , its Euclidean distance to y_i must satisfy:

$$\frac{1}{d}\|x - y_i\|_2^2 \leq D \quad \Longleftrightarrow \quad \|x - y_i\|_2 \leq \sqrt{dD}. \quad (20)$$

Hence, each codeword $y_i \in \mathcal{C}$ can cover at most a d -dimensional Euclidean ball $B_2^d(y_i, \sqrt{dD})$ of radius $r_{\text{code}} = \sqrt{dD}$.

Step 3 (Volume Ratio Lower Bound): Recall that the volume of a d -dimensional Euclidean ball of radius r is $\text{Vol}(B_2^d(r)) = c_d r^d$, where $c_d = \frac{\pi^{d/2}}{\Gamma(d/2+1)}$ depends only on d . For the union of K balls of radius $\sqrt{dD}$ to cover the source ball of radius $\sigma\sqrt{d}$, the sum of their volumes must be at least the volume of the source ball:

$$K \cdot \text{Vol}(B_2^d(\sqrt{dD})) \geq \text{Vol}(B_2^d(\sigma\sqrt{d})) \implies K \geq \frac{c_d(\sigma\sqrt{d})^d}{c_d(\sqrt{dD})^d} = \left(\frac{\sigma\sqrt{d}}{\sqrt{dD}}\right)^d = \left(\frac{\sigma^2}{D}\right)^{d/2}. \quad (21)$$

Substituting $K = 2^{dR}$ and taking $\frac{1}{d} \log_2(\cdot)$ of both sides gives:

$$R = \frac{\log_2 K}{d} \geq \frac{1}{d} \log_2 \left(\left(\frac{\sigma^2}{D} \right)^{d/2} \right) = \frac{1}{2} \log_2 \left(\frac{\sigma^2}{D} \right) \quad \Longleftrightarrow \quad D \geq \sigma^2 2^{-2R}. \quad \square$$

Achievability as $d \rightarrow \infty$. While our volume ratio gives a lower bound on K , random coding [9] shows that as $d \rightarrow \infty$, $K \approx (\sigma^2/D)^{d/2}$ randomly placed codewords on the sphere do cover almost all of the Gaussian measure within distance $\sqrt{dD}$. For example, at $R = 1$ bit and $\sigma^2 = 1$, the fundamental limit is $D^*(1) = 2^{-2} = 0.25$, and d -dimensional vector quantization converges to 0.25 as d increases ($d = 1 \implies 0.363$, $d = 2 \implies 0.36$, $d = 4 \implies 0.33$, $d = 8 \implies 0.29$, $d \rightarrow \infty \implies 0.25$).

6 Optimality of k -Means (Lloyd-Max) for Any Distribution

How do we actually construct an optimal K -point vector quantizer in $\mathbb{R}^d$ for an **arbitrary** probability distribution P (whether its coordinates are independent or strongly dependent)?

Given a random vector $X \sim P$ on $\mathbb{R}^d$ and codebook size K , we want to jointly choose a partition $\{S_1, \dots, S_K\}$ of $\mathbb{R}^d$ and codewords $\mathcal{C} = \{y_1, \dots, y_K\} \subset \mathbb{R}^d$ to minimize the expected squared Euclidean

error:

$$\mathcal{L}(\{S_i\}_{i=1}^K, \{y_i\}_{i=1}^K) \triangleq \mathbb{E}_X[\|X - Q(X)\|_2^2] = \sum_{i=1}^K \int_{S_i} \|x - y_i\|_2^2 dP(x). \quad (22)$$

This optimization is identical to the ***k*-Means clustering** problem in machine learning (originally developed for quantization as the **Lloyd-Max** or **Generalized Lloyd Algorithm** [6, 8]).

Theorem 6.1 (Necessary Optimality Conditions & Monotone Convergence of *k*-Means [6, 8]). *Let $X \sim P$ be any random vector on $\mathbb{R}^d$ with $\mathbb{E}[\|X\|_2^2] < \infty$ (with arbitrary dependent or independent coordinates). Then:*

1. **Optimal Partition for a Fixed Codebook (Voronoi Condition):** *For any fixed codewords $\mathcal{C} = \{y_1, \dots, y_K\}$, $\mathcal{L}(\{S_i\}, \{y_i\})$ is minimized by assigning every point $x \in \mathbb{R}^d$ to its nearest codeword in Euclidean distance:*

$$S_i^* = \left\{ x \in \mathbb{R}^d : \|x - y_i\|_2^2 \leq \|x - y_j\|_2^2 \ \forall j \in \{1, \dots, K\} \right\} \quad (\text{breaking ties arbitrarily}). \quad (23)$$

2. **Optimal Codebook for a Fixed Partition (Centroid Condition):** *For any fixed partition $\{S_1, \dots, S_K\}$ with $\Pr[X \in S_i] > 0$, $\mathcal{L}(\{S_i\}, \{y_i\})$ is **uniquely** minimized by setting each codeword y_i to the conditional expectation (center of mass) of its region:*

$$y_i^* = \mu_i \triangleq \mathbb{E}[X \mid X \in S_i] = \frac{\int_{S_i} x dP(x)}{\Pr[X \in S_i]}. \quad (24)$$

3. **Monotone Convergence:** *Alternating between (23) and (24) monotonically decreases $\mathcal{L}$ at every iteration and is guaranteed to converge for any distribution P .*

Proof. Step 1 (Proof of Voronoi Partition Optimality): Fix the codewords $y_1, \dots, y_K \in \mathbb{R}^d$. For any partition $\{S_1, \dots, S_K\}$ and every $x \in S_i$, we have $\|x - y_i\|_2^2 \geq \min_{j \in [K]} \|x - y_j\|_2^2$ pointwise. Integrating over P and summing over $i = 1, \dots, K$ yields:

$$\mathcal{L}(\{S_i\}, \{y_i\}) = \sum_{i=1}^K \int_{S_i} \|x - y_i\|_2^2 dP(x) \geq \sum_{i=1}^K \int_{S_i} \left(\min_{j \in [K]} \|x - y_j\|_2^2 \right) dP(x) = \int_{\mathbb{R}^d} \min_{j \in [K]} \|x - y_j\|_2^2 dP(x).$$

The lower bound on the right-hand side is achieved with equality if and only if $Q(x) = y_i$ satisfies $\|x - y_i\|_2^2 = \min_{j \in [K]} \|x - y_j\|_2^2$ for P -almost every x , which is precisely the Voronoi partition (23).

Step 2 (Proof of Centroid Optimality via Bias-Variance Decomposition): Now fix the partition $\{S_1, \dots, S_K\}$. Because the objective $\mathcal{L} = \sum_{i=1}^K \Pr[X \in S_i] \mathbb{E}[\|X - y_i\|_2^2 \mid X \in S_i]$ decouples across the K regions, we can minimize each i -th term independently over $y_i \in \mathbb{R}^d$. Let $\mu_i = \mathbb{E}[X \mid X \in S_i]$. Adding and subtracting μ_i inside the squared Euclidean norm:

$$\begin{aligned} \mathbb{E}[\|X - y_i\|_2^2 \mid X \in S_i] &= \mathbb{E}[\|(X - \mu_i) + (\mu_i - y_i)\|_2^2 \mid X \in S_i] \\ &= \mathbb{E}[\|X - \mu_i\|_2^2 \mid X \in S_i] + 2 \underbrace{\left\langle \mathbb{E}[X - \mu_i \mid X \in S_i], \mu_i - y_i \right\rangle}_{= \mu_i - \mu_i = 0} + \|\mu_i - y_i\|_2^2 \\ &= \mathbb{E}[\|X - \mu_i\|_2^2 \mid X \in S_i] + \|\mu_i - y_i\|_2^2. \end{aligned} \quad (25)$$

The first term is the conditional variance of X on S_i (which does not depend on y_i), and the second term satisfies $\|\mu_i - y_i\|_2^2 \geq 0$ with equality **if and only if** $y_i = \mu_i$. Thus $y_i^* = \mathbb{E}[X \mid X \in S_i]$ is the unique optimal codeword for region S_i .

Step 3 (Monotone Convergence of Alternating Minimization): Starting from an initial codebook $y^{(0)}$, let $S^{(t)}$ be the Voronoi partition for $y^{(t-1)}$ and let $y^{(t)}$ be the conditional means for $S^{(t)}$. By Steps 1 and 2:

$$\mathcal{L}(S^{(t)}, y^{(t-1)}) \geq \mathcal{L}(S^{(t)}, y^{(t)}) \geq \mathcal{L}(S^{(t+1)}, y^{(t)}) \geq 0.$$

Because the sequence of distortion values is monotonically non-increasing and bounded below by 0, it must converge to a limit $\mathcal{L}^* \geq 0$. $\square$

7 The Curse of Dictionary Size & Product Quantization (PQ)

7.1 Why Full-Dimensional k -Means Is Impossible in High Dimensions

Theorem 6.1 tells us that k -means finds locally optimal vector quantizers in $\mathbb{R}^d$. Why can't we simply run k -means directly on d -dimensional LLM embeddings or KV-cache vectors?

Look at the relationship between the per-coordinate rate R , the dimension d , and the dictionary size K :

$$K = 2^{dR}. \quad (26)$$

- Suppose $d = 128$ (a single attention head in Llama-3) and we want a modest rate of $R = 2$ bits per float.
- Then the codebook must contain $K = 2^{128 \times 2} = 2^{256} \approx 1.15 \times 10^{77}$ centroids—roughly the number of atoms in the observable universe!
- Even if $d = 64$ and $R = 1$ bit/float, $K = 2^{64} \approx 1.8 \times 10^{19}$. Both **storing the dictionary** ($\Theta(Kd)$ floats) and **encoding a vector** by searching for its nearest centroid ($\Theta(Kd)$ operations) grow **exponentially** in d , making direct d -dimensional VQ completely intractable.

7.2 Product Quantization (PQ — Jégou, Douze, & Schmid, 2011)

To get an exponentially large codebook of size $K = 2^{dR}$ **without** exponential storage or search cost, **Product Quantization (PQ)** [5] decomposes $\mathbb{R}^d$ into m smaller subspaces of dimension $d' = d/m$ and quantizes each subspace using a small k -means codebook.

Definition 7.1 (Product Quantizer [5]). *Partition $x \in \mathbb{R}^d$ into m contiguous subvectors $x = (x^{(1)}, x^{(2)}, \dots, x^{(m)})$ where each $x^{(j)} \in \mathbb{R}^{d'}$ has dimension $d' = d/m$. Let $Q_j : \mathbb{R}^{d'} \rightarrow \mathcal{C}_j$ be a d' -dimensional vector quantizer on the j -th subspace with codebook $\mathcal{C}_j = \{y_1^{(j)}, \dots, y_k^{(j)}\} \subset \mathbb{R}^{d'}$ of size $k = 2^{d'R}$. The **Product Quantizer** $Q_{PQ} : \mathbb{R}^d \rightarrow \mathcal{C}_1 \times \dots \times \mathcal{C}_m$ maps x to the concatenation of its m quantized subvectors:*

$$Q_{PQ}(x) \triangleq (Q_1(x^{(1)}), Q_2(x^{(2)}), \dots, Q_m(x^{(m)})). \quad (27)$$

Why Product Quantization Breaks the Curse of Dictionary Size.

1. **Exponential Effective Codebook Size** ($k^m = 2^{dR}$): Because each of the m subvectors can independently choose any of k sub-centroids, the implicit global codebook is the **Cartesian product** $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \cdots \times \mathcal{C}_m$, which contains:

$$K = |\mathcal{C}| = k^m = (2^{d'R})^m = 2^{(md')R} = 2^{dR} \quad \text{distinct codewords in } \mathbb{R}^d.$$

Each vector is stored as m small integers $(i_1, \dots, i_m) \in \{1, \dots, k\}^m$, taking $m \log_2 k = dR$ bits.

2. **Tiny Dictionary Storage & Encoding Cost** ($\mathcal{O}(kd)$): Instead of storing $K = k^m$ vectors of dimension d , we only store m sub-codebooks of k vectors of dimension $d' = d/m$, which takes:

$$\text{Dictionary Floats} = m \cdot k \cdot \frac{d}{m} = kd \quad \text{floats,}$$

and quantizing a vector $x \in \mathbb{R}^d$ requires finding the nearest of k centroids in each of the m subspaces, taking $m \cdot \mathcal{O}(kd') = \mathcal{O}(kd)$ time. (*Example: For $d = 128$, $R = 2$ bits/coord, choosing $m = 32$ blocks of dimension $d' = 4$ gives $k = 2^{4 \times 2} = 256$ centroids per block. We represent $K = 256^{32} = 2^{256}$ codewords while storing just $kd = 256 \times 128 = 32,768$ floats.*)

3. **Fast Lookup-Table Inner Products for KV-Cache & Vector Search**: To compute the inner product $\langle q, Q_{\text{PQ}}(x) \rangle$ between a query $q = (q^{(1)}, \dots, q^{(m)}) \in \mathbb{R}^d$ and N quantized vectors, we first precompute an $m \times k$ lookup table $T[j, \ell] = \langle q^{(j)}, y_\ell^{(j)} \rangle$ in $\mathcal{O}(kd)$ time once per query, and then evaluate each vector's inner product in just m table lookups and additions: $\langle q, Q_{\text{PQ}}(x) \rangle = \sum_{j=1}^m T[j, i_j]$.

Quantization Paradigm	Block d'	Blocks m	Sub-Codebook k	Codewords K	Storage & Encode
Scalar Quantization (SQ)	$d' = 1$	$m = d$	$k = 2^R$	2^{dR}	$\Theta(2^R \cdot d)$
Product Quantization (PQ)	$d' = d/m$	$m = d/d'$	$k = 2^{d'R}$	$k^m = 2^{dR}$	$\Theta(2^{(d/m)R} \cdot d)$
Full Vector Quantization	$d' = d$	$m = 1$	$k = 2^{dR}$	2^{dR}	$\Theta(2^{dR} \cdot d)$ (intractable)

8 Zador's Scaling & Proving That SVD Provably Improves PQ

Because Product Quantization quantizes the m coordinate groups $X^{(1)}, \dots, X^{(m)}$ **independently**, its distortion depends heavily on the coordinate variances in each group. Suppose our data vectors $X \in \mathbb{R}^d$ have a symmetric positive semi-definite (PSD) covariance matrix $\Sigma = \mathbb{E}[XX^T] \succeq 0$. If we pre-multiply X by a rotation matrix $R \in \mathbb{R}^{d \times d}$ before applying Product Quantization, the rotated vector $\tilde{X} = RX$ has covariance matrix:

$$\tilde{\Sigma} = \mathbb{E}[(RX)(RX)^T] = R\Sigma R^T, \quad \text{with diagonal variances } \sigma_i^2 \triangleq (R\Sigma R^T)_{ii}.$$

Which rotation R minimizes the Product Quantization distortion? We now build a step-by-step deductive proof showing that **it is always optimal to SVD the matrix Σ first and assign the principal coordinates to the m groups of size $d' = d/m$.**

8.1 Step 1: Zador’s Lemma in Terms of σ_i and the PQ Cost Function

We begin with a fundamental result from high-rate quantization theory—originally established by **Zador** [11] and **Gersho** [3] (see also **Gersho & Gray** [4] and **Ge et al.** [2])—which states how the k -means distortion scales with the coordinate standard deviations $\sigma_1, \dots, \sigma_d$ (stated here without proof):

Lemma 8.1 (Zador’s Scaling of k -Means Distortion [11, 3]). *Let $X \in \mathbb{R}^d$ be a d -dimensional Gaussian vector with coordinate standard deviations $\sigma_1, \dots, \sigma_d > 0$ (variances $\sigma_1^2, \dots, \sigma_d^2$). For any fixed dimension d and fixed number of centroids k , the optimal k -means MSE distortion is proportional to $(\prod_{i=1}^d \sigma_i)^{2/d} = (\prod_{i=1}^d \sigma_i^2)^{1/d}$:*

$$D_{VQ}(\sigma_1, \dots, \sigma_d) = C_{d,k} \left(\prod_{i=1}^d \sigma_i \right)^{2/d} = C_{d,k} \left(\prod_{i=1}^d \sigma_i^2 \right)^{1/d}, \quad (28)$$

where $C_{d,k} > 0$ is a constant depending only on d and k (for $k = 2^{dR}$, $C_{d,k} \propto k^{-2/d} = 2^{-2R}$).

Geometric Intuition (From High-Density Volume to Quadratic Distortion). Why is the distortion proportional to $(\prod_{i=1}^d \sigma_i)^{2/d}$?

1. **Volume is $\prod_{i=1}^d \sigma_i$:** A d -dimensional distribution with coordinate standard deviations $\sigma_1, \dots, \sigma_d$ concentrates its probability mass inside an axis-aligned box/ellipsoid whose d -dimensional volume is obviously proportional to the product of the side lengths: $\text{Vol} \propto \prod_{i=1}^d \sigma_i$.
2. **Distortion is Quadratic in the Cell Radius (As Seen in Section 5):** As we saw earlier in our sphere-covering proof (Theorem 5.1), when we partition a region of volume Vol into k cells of radius r , each cell has volume proportional to r^d , so $k \cdot r^d \propto \text{Vol} \propto \prod_{i=1}^d \sigma_i$. Solving for the cell radius gives $r \propto (\prod_{i=1}^d \sigma_i)^{1/d}$ (for fixed k and d). Because squared Euclidean distortion (MSE) is **quadratic** in the radius r , the distortion scales as:

$$D_{VQ} \propto r^2 \propto \left(\prod_{i=1}^d \sigma_i \right)^{2/d} = \left(\prod_{i=1}^d \sigma_i^2 \right)^{1/d}.$$

Consequence for Product Quantization (PQ Cost Function). In Product Quantization, we partition the d coordinates into m disjoint groups $B_1, \dots, B_m$, each of size $d' = d/m$, and quantize each group with k centroids. Since the total squared Euclidean error is the sum of the squared errors across the m groups, applying Lemma 8.1 to each group B_j gives the **PQ cost function** in terms of the diagonal variances $\sigma^2 = (\sigma_1^2, \dots, \sigma_d^2)^T$:

$$D_{PQ}(\sigma^2) = C_{d',k} \sum_{j=1}^m \left(\prod_{i \in B_j} \sigma_i^2 \right)^{1/d'}, \quad \text{where } \sigma_i^2 = (R \Sigma R^T)_{ii}. \quad (29)$$

8.2 Step 2: Reminder — Convex Sets, Concave Functions, and Extreme Points

To minimize $D_{PQ}(\sigma^2)$ over all rotations R , let us first recall three basic definitions from convex analysis:

Definition 8.2 (Convex Set, Concave Function, and Extreme Point).

1. **Convex Set:** A set $\mathcal{K} \subseteq \mathbb{R}^d$ is **convex** if for any two points $u, v \in \mathcal{K}$ and any $\theta \in [0, 1]$, the line segment between them lies inside $\mathcal{K}$:

$$\theta u + (1 - \theta)v \in \mathcal{K}.$$

2. **Concave Function (and by Negation, Convex Function):** A function $f : \mathcal{K} \rightarrow \mathbb{R}$ defined on a convex set $\mathcal{K}$ is **concave** if for all $u, v \in \mathcal{K}$ and all $\theta \in [0, 1]$:

$$f(\theta u + (1 - \theta)v) \geq \theta f(u) + (1 - \theta)f(v). \quad (30)$$

(By negation, $-f$ is **convex**, meaning $(-f)(\theta u + (1 - \theta)v) \leq \theta(-f)(u) + (1 - \theta)(-f)(v)$.)

3. **Extreme Point (Vertex):** A point $z \in \mathcal{K}$ is an **extreme point** of $\mathcal{K}$ if it cannot be written as a strict convex combination $z = \theta u + (1 - \theta)v$ of two distinct points $u \neq v$ in $\mathcal{K}$ with $\theta \in (0, 1)$. For a convex polytope, the extreme points are simply its **vertices (corners)**.

Why is concavity on a convex set so useful for minimization? Because a concave function always curves *above* the secant line connecting its endpoints—meaning it can never dip strictly below both endpoints in the interior of a line segment! Repeating this argument across faces of a polytope yields a classic theorem (stated here without proof):

Fact 8.3 (Minimum of a Concave Function Is Attained at an Extreme Point). *Let $\mathcal{K} \subset \mathbb{R}^d$ be a compact convex polytope and let $f : \mathcal{K} \rightarrow \mathbb{R}$ be a continuous **concave** function on $\mathcal{K}$ (equivalently, $-f$ is convex). Then f attains its **global minimum** over $\mathcal{K}$ at an **extreme point (vertex)** of $\mathcal{K}$:*

$$\min_{v \in \mathcal{K}} f(v) = \min_{v \in \text{Ext}(\mathcal{K})} f(v). \quad (31)$$

Our Deductive Strategy. To apply Fact 8.3 to find the best rotation R , we simply need to establish two ingredients:

1. **The Convex Set:** Show that as R varies, the vector of diagonal entries $\sigma^2 = \text{diag}(R\Sigma R^T)$ lies inside a convex polytope—the **majorization polytope**—whose extreme points are permutations of the eigenvalues of Σ .
2. **The Concave Function:** Show that the PQ cost function $D_{\text{PQ}}(\sigma^2)$ in (29) is **concave** with respect to the diagonal values σ^2 .

8.3 Step 3: The Convex Set — The Majorization Polytope and the Diagonal of $R\Sigma R^T$

Definition 8.4 (Majorization and the Majorization Polytope [7]). *Given a vector $\lambda = (\lambda_1, \dots, \lambda_d)^T \in \mathbb{R}_{\geq 0}^d$ sorted in non-increasing order ($\lambda_1 \geq \dots \geq \lambda_d \geq 0$), a vector $v \in \mathbb{R}_{\geq 0}^d$ is **majorized** by λ (written $v \prec \lambda$) if for every $t \in \{1, \dots, d\}$, the sum of any t entries of v is at most the sum of the t largest entries of λ :*

$$\sum_{i \in T} v_i \leq \sum_{i=1}^t \lambda_i \quad \text{for all } T \subseteq [d], |T| = t, \quad \text{with total sum } \sum_{i=1}^d v_i = \sum_{i=1}^d \lambda_i. \quad (32)$$

Because (32) is defined by a finite collection of linear inequalities, the set of all vectors majorized by λ is **trivially a convex polytope**—called the **majorization polytope** $\mathcal{P}(\lambda) = \text{conv}\{P\lambda : P \text{ is a permutation matrix}\}$ —whose **extreme points (vertices)** are the permutations $P\lambda$ of λ .

Now let $\Sigma \in \mathbb{R}^{d \times d}$ be a symmetric positive semi-definite (PSD) matrix with eigendecomposition (SVD):

$$\Sigma = U\Lambda U^T, \quad \text{where } U^T U = U U^T = I_d \text{ and } \Lambda = \text{diag}(\lambda_1, \dots, \lambda_d) \text{ with } \lambda_1 \geq \dots \geq \lambda_d \geq 0,$$

and let $\lambda = \text{diag}(\Lambda) = (\lambda_1, \dots, \lambda_d)^T \in \mathbb{R}^d$ be the vector of eigenvalues of Σ . We show that for any matrix R with $\|R\|_2 \leq 1$, the diagonal entries of $R\Sigma R^T$ lie in the majorization polytope $\mathcal{P}(\lambda)$:

Lemma 8.5 (Diagonal Entries of $R\Sigma R^T$ Lie in the Majorization Polytope). *Let $\Sigma = U\Lambda U^T \succeq 0$ be symmetric PSD and let $\|R\|_2 \leq 1$ (note that the set $\{R : \|R\|_2 \leq 1\}$ is trivially convex, and clearly no optimal solution has $\|R\|_2 < 1$ strictly smaller than 1). Then the vector of diagonal entries $\sigma^2 = \text{diag}(R\Sigma R^T) \in \mathbb{R}^d$ lies in the majorization polytope $\mathcal{P}(\lambda)$.*

Proof. Write $\Sigma = U\Lambda U^T$. Absorbing the orthogonal matrix U into $\tilde{R} \triangleq RU$ preserves $\|\tilde{R}\|_2 = \|R\|_2 \leq 1$ and gives $R\Sigma R^T = \tilde{R}\Lambda\tilde{R}^T$, so it suffices to consider $\text{diag}(\tilde{R}\Lambda\tilde{R}^T)$ for the diagonal matrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_d)$.

For any subset $T \subseteq [d]$ of t coordinates, let $\tilde{R}_T \in \mathbb{R}^{t \times d}$ denote the submatrix of the t rows of $\tilde{R}$ in T . Because $\|\tilde{R}_T\|_2 \leq \|\tilde{R}\|_2 \leq 1$, the sum of the t diagonal entries in T cannot exceed the sum of the t largest entries of Λ :

$$\sum_{i \in T} (\tilde{R}\Lambda\tilde{R}^T)_{ii} = \text{Tr}(\tilde{R}_T\Lambda\tilde{R}_T^T) \leq \sum_{i=1}^t \lambda_i. \quad (33)$$

In other words, **no** $\text{diag}(\tilde{R}\Lambda\tilde{R}^T)$ **can majorize** $\text{diag}(\Lambda)$ —every $\sigma^2 = \text{diag}(R\Sigma R^T)$ is majorized by $\lambda = \text{diag}(\Lambda)$ and therefore lies in the majorization polytope $\mathcal{P}(\lambda)$. $\square$

8.4 Step 4: The Concave Function — Concavity of the PQ Cost w.r.t. the Diagonal Values

Next, we check the second ingredient required by Fact 8.3: that the PQ cost function $D_{\text{PQ}}(\sigma^2)$ is a **concave function** of the diagonal variance vector $v = \sigma^2 = (\sigma_1^2, \dots, \sigma_d^2)^T \in \mathbb{R}_{\geq 0}^d$.

Lemma 8.6 (Concavity of the PQ Cost Function). *As a function of the diagonal values $v = \sigma^2 = (\sigma_1^2, \dots, \sigma_d^2)^T \in \mathbb{R}_{\geq 0}^d$, the Product Quantization distortion*

$$D_{\text{PQ}}(v) = C_{d',k} \sum_{j=1}^m \left(\prod_{i \in B_j} v_i \right)^{1/d'}$$

is concave on $\mathbb{R}_{\geq 0}^d$.

Proof. Consider a single group B_j of size $d' = d/m$, and its cost term $g(v_{B_j}) = (\prod_{i \in B_j} v_i)^{1/d'}$ (the **geometric mean** of the d' diagonal entries in group B_j). For any two non-negative vectors $u, w \in \mathbb{R}_{\geq 0}^{d'}$

and any $\theta \in [0, 1]$, applying **Hölder's inequality** with exponents $p_i = d'$ (so $\sum_{i \in B_j} \frac{1}{d'} = 1$) gives:

$$g(\theta u + (1 - \theta)w) = \prod_{i \in B_j} (\theta u_i + (1 - \theta)w_i)^{1/d'} \geq \theta \prod_{i \in B_j} u_i^{1/d'} + (1 - \theta) \prod_{i \in B_j} w_i^{1/d'} = \theta g(u) + (1 - \theta)g(w).$$

Thus each group's geometric mean $(\prod_{i \in B_j} v_i)^{1/d'}$ is concave. Since a sum of concave functions multiplied by $C_{d',k} > 0$ is also concave, $D_{\text{PQ}}(v)$ is concave on $\mathbb{R}_{\geq 0}^d$. $\square$

8.5 Step 5: Putting It All Together — Optimality of SVD ($R = PU^T$) and Optimized Product Quantization (OPQ)

We now combine Fact 8.3, Lemma 8.5, and Lemma 8.6 to immediately deduce the optimal rotation R for Product Quantization—recovering the theoretical foundation of **Optimized Product Quantization (OPQ)** [2]:

Theorem 8.7 (Optimality of SVD for Product Quantization [2]). *Let $\Sigma = U\Lambda U^T \succeq 0$ be the SVD of the covariance matrix Σ , where U are the left singular vectors (eigenvectors) of Σ and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_d)$. Over all rotations $R \in \mathbb{R}^{d \times d}$ ($\|R\|_2 \leq 1$, $RR^T = I_d$), the Product Quantization cost $D_{\text{PQ}}(\text{diag}(R\Sigma R^T))$ attains its **minimum** at an extreme point $\sigma^2 = P\lambda$ of the majorization polytope $\mathcal{P}(\lambda)$, which is achieved by:*

$$R^* = PU^T, \tag{34}$$

where $P \in \{0, 1\}^{d \times d}$ is a permutation matrix.

Proof. 1. By **Lemma 8.5**, for any rotation R , the vector of diagonal variances $\sigma^2 = \text{diag}(R\Sigma R^T)$ belongs to the majorization polytope $\mathcal{P}(\lambda)$ —a compact **convex set** whose extreme points are the permuted eigenvalue vectors $P\lambda$.

2. By **Lemma 8.6**, the PQ distortion $D_{\text{PQ}}(\sigma^2)$ is a **concave function** of σ^2 .

3. By **Fact 8.3**, the minimum of the concave function $D_{\text{PQ}}(\sigma^2)$ over the convex polytope $\mathcal{P}(\lambda)$ is attained at an **extreme point** of $\mathcal{P}(\lambda)$, namely:

$$\sigma^2 = P\lambda = (\lambda_{\pi(1)}, \dots, \lambda_{\pi(d)})^T$$

for some permutation matrix P .

4. Finally, which rotation R realizes the extreme point $\sigma^2 = P\lambda$? Setting $\tilde{R} = RU = P$ and multiplying on the right by U^T gives:

$$R^* = PU^T.$$

Indeed, substituting $R^* = PU^T$ gives $R^*\Sigma(R^*)^T = PU^T U \Lambda U^T U P^T = P \Lambda P^T = \text{diag}(\lambda_{\pi(1)}, \dots, \lambda_{\pi(d)})$, so $\text{diag}(R^*\Sigma(R^*)^T) = P\lambda$, completing the proof. $\square$

Conclusion & Practical Recipe — Optimized Product Quantization (OPQ [2]). Theorem 8.7 shows that **it is always better to SVD the covariance matrix $\Sigma = U\Lambda U^T$ first** and then assign the principal coordinates to groups of size $d' = d/m$. This procedure is known in the literature as **Optimized Product Quantization (OPQ)** [2]:

1. **SVD Alignment (U^T):** Rotate every vector $x \mapsto U^T x$ into the SVD principal component basis of Σ , decorrelating the coordinates so that their variances are the singular values $\lambda_1, \dots, \lambda_d$.
2. **Eigenvalue Allocation / Permutation (P):** Permute the d principal coordinates into the m groups $B_1, \dots, B_m$ of size $d' = d/m$ (e.g., greedily assigning each eigenvalue λ_i to the group with the smallest running product $\prod_{\ell \in B_j} \lambda_\ell$ so the group geometric means are balanced) before running k -means on each group.

Summary Table

Topic / Method	Mathematical Mechanism	Codebook / Complexity	Guarantee / Key Property
1.2 Uniform 1D Quantizer	$K = 2^R$ intervals of step $\Delta = 2a/2^R$	$K = 2^R$ scalars	$D(R) = \frac{\Delta^2}{12} = \frac{a^2}{3} 2^{-2R}$
1.2 Gaussian 1-Bit Quantizer	$Q(x) = \text{sign}(x) \cdot \mathbb{E}[X]$	$K = 2$ scalars ($\pm\sigma\sqrt{2/\pi}$)	$D_{\text{SQ}}(1) = (1 - \frac{2}{\pi})\sigma^2 \approx 0.3634\sigma^2$
1.3 Rotated Scalar Quantizer [10, 1, 12]	Rotate $\tilde{x} = Ux \approx \mathcal{N}(0, \frac{\ x\ _2^2}{d} I_d) + 1\text{D SQ}$	2^R scalars, $\mathcal{O}(d \log d)$ time	Eliminates outliers across all coordinates
1.4 2D Hexagonal Lattice VQ	Tile $\mathbb{R}^2$ with hexagons of area $A = \frac{3\sqrt{3}}{2}\delta^2$	$K = 2^{2R}$ 2D codewords	$\frac{\text{MSE}_{\text{hex}}}{\text{MSE}_{\text{square}}} = \frac{5}{3\sqrt{3}} \approx 0.962$ (3.8% gain on i.i.d. uniform)
2.1 Gaussian R-D Limit [9]	Cover $B_2^d(\sigma\sqrt{d})$ with 2^{dR} balls $B_2^d(\sqrt{dD})$	$K = 2^{dR}$ d -D codewords	$R(D) \geq \frac{1}{2} \log_2(\sigma^2/D) \iff D^*(R) = \sigma^2 2^{-2R}$
2.2 k-Means (Lloyd-Max) [6, 8]	Alternating Voronoi cells & $y_i^* = \mathbb{E}[X \mid X \in S_i]$	$K = 2^{dR}$ centroids ($\Theta(Kd)$)	Necessary optimality & monotone convergence for any P
2.3 Product Quantization (PQ) [5]	Cartesian product $\mathcal{C}_1 \times \dots \times \mathcal{C}_m$ on $\mathbb{R}^{d/m}$	mk centroids stored, $k^m = 2^{dR}$ total	$\mathcal{O}(kd)$ space/time; $D_{\text{PQ}}(\sigma^2) = C_{d',k} \sum_{j=1}^m (\prod_{i \in B_j} \sigma_i^2)^{1/d'}$
2.4 SVD + PQ (OPQ) [2]	Concave $D_{\text{PQ}}(\sigma^2)$ over majorization polytope $\mathcal{P}(\lambda)$	$\mathcal{O}(d^2)$ rotation + $\mathcal{O}(kd)$ PQ	Minimum at extreme point $\sigma^2 = P\lambda$ via $R^* = PU^T$

References

- [1] Jianyang Gao and Cheng Long. RaBitQ: Quantizing high-dimensional vectors with a theoretical error bound for approximate nearest neighbor search. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2024.
- [2] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):744–755, 2013.
- [3] Allen Gersho. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, 25(4):373–380, 1979.
- [4] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Springer, 1992.
- [5] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.
- [6] Stuart P. Lloyd. Least squares quantization in PCM. *Bell Telephone Laboratories Paper*, 1957. (Published in *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.)

- [7] Albert W. Marshall, Ingram Olkin, and Barry C. Arnold. *Inequalities: Theory of Majorization and Its Applications*. Springer, 2nd edition, 2011.
- [8] Joel Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6(1):7–12, 1960.
- [9] Claude E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE National Convention Record*, Part 4, pages 142–163, 1959.
- [10] Shay Vargaftik, Ran Ben Basat, Amit Portnoy, Gil Einziger, Roy Friedman, and Michael Mitzenmacher. EDEN: Communication-efficient and robust distributed mean estimation for federated learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 21984–22014, 2022.
- [11] Paul L. Zador. *Development and Evaluation of Procedures for Quantizing Multivariate Distributions*. Ph.D. Dissertation, Stanford University, 1963. (Published as “Asymptotic quantization error of continuous signals and the quantization dimension,” *IEEE Transactions on Information Theory*, 28(2):139–149, 1982.)
- [12] Amir Zandieh, Majid Daliri, and Insu Han. TurboQuant: Online vector quantization with near-optimal distortion rate. *arXiv preprint arXiv:2504.19874*, 2025.