# Reusable Codebooks: Hierarchical TurboQuant Quantization for Two-Tier Approximate Nearest-Neighbor Retrieval

Shashwat Saket
ss20308@nyu.edu
New York University
Under Prof. Ainesh Bakshi

May 2026

**Abstract**

The hierarchical codebooks generation for vector quantization can be approximated during the retrieval of approximate nearest neighbor. TurboQuant has data oblivious Lloyd Max Gaussian quantization on top of which this paper proposes three different constructions of nested codebooks — bottom-up, top-down, middle-anchor — generating codebooks for multi bit-width $\{b_1 < b_2 < \cdots < b_{\max}\}$ such that the codebook of higher bit-width can be formed using codebook of lower bit-width. This enables us to have a shared codebook for the prefilter (which can be done using lower bits) and the reranker (which is expensive and can be done using higher bit quantization) in a two-tier retrieval pipeline. This nesting property removes the training and storage overhead for the independent per-tier codebook. The MSE penalty for each construction is characterized relative to the unconstrained Lloyd-Max codebooks, and empirically it is visible that the top-down decimation matches the unconstrained Lloyd-Max within a limited range of statistical noise for the expensive tier. This paper finds that the nested top-down TurboQuant matches the standard Product Quantization + reranking baseline in retrieval accuracy in high dimensions and surpasses it by +2.2 R@1 for lower dimensions on GloVe-200, OpenAI-1536, and OpenAI-3072 datasets, while requiring only one shared codebook (vs. two independent PQ codebooks) and offering 50–1300× faster index construction.

## 1 Introduction

For retrieving the vectors, the standard practice is to use an expanding production system which includes semantic search, RAG, recommendation, and matching. As the size of the corpus scales and becomes more than thousands of millions of high dimensional embeddings, the full nearest neighbor search becomes restrictive, because of which a quantized ANN is required to address this by encoding the vectors using a limited number of bits and getting the approximate inner product directly on the codes.

In production, a two-tier approach is implemented for retrieval which includes a cheap lower-bit codebook and an expensive higher-bit codebook, where the lower bit is used to score all $N$ vectors from the database for a given query, and then the higher bit is used to rerank the selected candidates for getting the top-$k$ results. With this pipeline, it is modular to get the cost of full database scan and the cost of accuracy on the scores.

The standard implementation of this approach involves two different Product Quantization (PQ) codebooks trained at separate bit widths. Because of this, there is double codebook storage with two separate $k$-means runs, and code stored per vector for both the bit widths.

Here the optimal scalar codebook at $2^{b_{\text{cheap}}}$ levels — which is the Lloyd-Max-based codebook of the source distribution — is not usually a subset of the optimal codebook at the expensive $2^{b_{\text{exp}}}$ bit level. To force this kind of nesting, it is required to accept some mean-square-error penalty relative to the unconstrained Lloyd-Max codebooks. This paper addresses if this penalty can be measured in retrieval accuracy loss.

### 1.1 Contributions

This paper makes the following contributions:

1. **Construction of three different strategies of nested codebooks.** Using TurboQuant's [**?**] Lloyd-Max-Gaussian quantizer, the bottom-up, top-down, and middle-anchor codebooks have been constructed, and for each construction, MSE penalty relative to unconstrained Lloyd-Max codebooks have been characterized.
2. **A two-tier retrieval pipeline** (Section 6). The top-down approach for the nested codebooks fosters a cheap-tier-based prefiltering and then the actual expensive-tier reranking using a single shared codebook.
3. **Theoretical foundations** (Section 5). We prove three results that explain the empirical observations: a Conservation Theorem showing no nested family can achieve zero MSE penalty at two distinct bit-widths simultaneously; a Monotone Penalty Theorem characterizing why top-down decimation penalty grows with $b_{\max} - b$; and a Retrieval-MSE Divergence Theorem explaining why MSE-optimal placement is not in general retrieval-optimal.
4. **Empirically comprehensive evaluation** (Section 7). On three different real world datasets — GloVe-200, OpenAI-1536, OpenAI-3072 — with full query count which is 10K queries for GloVe-200 and 1K queries for the OpenAI datasets, with 5-seed multi-seed validation and then comparing the results against PQ + reranking. This paper also demonstrates that the top-down nested TurboQuant is matching the unconstrained TurboQuant within $\pm 0.005$ R@1 for all the three datasets and also exceeds or matches PQ + reranking, which displays concrete advantages in storage and index construction time complexity.

## 2 Related Work

### 2.1 Lloyd-Max codebooks and scalar quantization

Lloyd [1] and Max [2] derived the conditions on MSE-optimal quantization, which is a foundational result on optimal scalar quantization: nearest-neighbor encoding (the partition condition) and centroid decoding (the centroid condition). The algorithm iterates through these two conditions to generate the optimal codebooks for a given dataset.

In the case of continuous source distributions, Bennett [3] and Gish & Pierce [4] prove that with increasing codebook size, as $n \to \infty$, levels of an optimal scalar quantizer asymptotically tend toward a point density proportional to $f(x)^{1/3}$, where $f$ is the source density. This connects the discrete optimization problem to continuous functional analysis and provides distortion-rate bounds.

### 2.2 Product Quantization and its variants

Product Quantization (PQ) [6] is an algorithm which divides vector space into $m$ different subspaces and trains a separate $2^b$-level codebook using $k$-means for each of the subspaces. In production systems, PQ has dominated ANN systems for over a decade and is considered a standard baseline. Optimized Product Quantization (OPQ) [7] is similar to PQ with an added rotation. Additive Quantization [8] uses the generalization strategy with additive structures, and Composite Quantization [9] adds further refinement.

The PQ family methods have a data-dependency characteristic, which means the codebook adapts to the empirical distribution of the dataset. This makes them expensive to compute in the case of high-dimensional, large-bit-width settings, and further has limitations which can cause overfitting in codebooks for heterogeneous corpora.

### 2.3 Data-oblivious quantization

A similar line of work has also been explored contemporarily which works on the principle of data-oblivious quantization. This means that once the codebook has been selected independent of the database, then a randomized rotation is applied uniformly which approximates Gaussianization of the rotated coordinates.

RaBitQ [10] implements 1-bit-per-coordinate sign quantization on randomly rotated unit-norm vectors, hereby achieving competitive recall via theoretical guarantees for rotation-induced concentration. Extended RaBitQ [12] extends this methodology to multi-bit codes having magnitude information stored, which gets strong recall at $b = 4$.

TurboQuant [**?**] generalizes this data-oblivious approach from RaBitQ to arbitrary bit-widths via Lloyd-Max-Gaussian quantization on standardized rotated coordinates. It has two variants: TurboQuant-MSE, which implements Lloyd-Max on rotated coordinates, and TurboQuant-Prod, implementing the bias-correction-based QJL on the residual of unbiased inner-product estimation. All the TurboQuant numbers used in this paper use the TurboQuant-MSE variant results.

### 2.4 Hierarchical retrieval and the two-tier architecture

The two-tier architecture, which uses cheap prefilter followed by expensive reranking, is a standard used in production-based ANN systems. IVFADC [6] and IVF-PQ are two canonical pipelines which use an inverted file index for generating the candidate clusters, then PQ codes rerank within each cluster. Specific to this paper, prefiltering uses a lower-bit PQ and reranking uses a higher-bit PQ with independent codebooks. There is no prior work which explored shared codebooks for two-tier retrieval based on nested data-oblivious scalar quantization.

### 2.5 Multi-resolution and progressive coding

Hierarchical coding in progressive transmission has a prelude in image and video compression [14], with tree-based vector quantization [13] which implements nested codebooks for fast search. These were used to optimize rate distortion and decoding latency, and operated on a different distribution and objective than ours.

## 3 Background: TurboQuant

### 3.1 The TurboQuant pipeline

For a dataset with $\mathbf{X} \in \mathbb{R}^{N \times d}$ and unit-norm query $q \in \mathbb{R}^d$:

**Setup (offline, once):**

1. Sample a Haar-distributed orthogonal matrix $R \in \mathbb{R}^{d \times d}$.
2. Compute the rotated database $\tilde{\mathbf{X}} = \mathbf{X}R^\top$.
3. Standardize: $\bar{\mathbf{X}} = \sqrt{d} \cdot \tilde{\mathbf{X}}$.
4. Compute the Lloyd-Max codebook for the standard Gaussian at $2^b$ levels.
5. Encode each row $\bar{x}_i$ coordinate-wise: $\text{code}_i[j] = \arg\min_l |\bar{x}_i[j] - L[l]|$.

**Online (per query):**

1. Rotate and standardize $q$: $\bar{q} = \sqrt{d} \cdot Rq$.
2. For each database vector $i$, reconstruct $\hat{x}_i[j] = L[\text{code}_i[j]]/\sqrt{d}$.
3. Score $s_i = \bar{q}^\top \hat{x}_i / d$.
4. Return top-$k$ using score.

Here $\bar{x}$ has approximately i.i.d. $\mathcal{N}(0, 1)$ coordinates, having a single Lloyd-Max-Gaussian codebook sufficient for every coordinate, every dataset, every dimension. Because of this, there is no requirement of data-dependent training.

### 3.2 Lloyd-Max-Gaussian codebooks

In the case of the standard Gaussian $f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$, the codebook $L_n^*$ can be computed with Lloyd iteration. For this, it is required to:

- Initialize: evenly spaced levels in $[-3, 3]$ which is the $3\sigma$ range.
- Iterate: alternately iterate between conditional-mean centroids of each partition and midpoint partition boundaries.
- Converge: usually converges in 50–100 iterations with $10^{-8}$ tolerance.

## 4 Hierarchical Codebook Constructions

A nested codebook family is a set $\{L_b\}_{b=b_1}^{b_{\max}}$ where $|L_b| = 2^b$ and $L_{b_i} \subset L_{b_j}$ for $i < j$. We propose three constructions for nested Lloyd-Max-Gaussian codebooks. All operate on the standard Gaussian source distribution $f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$.

### 4.1 Bottom-up construction

Anchor at $b_1$ (the smallest bit-width, optimal at that level), then expand to higher bit-widths via constrained Lloyd-Max.

**Algorithm 1** Nested Bottom-up Lloyd-Max codebook construction

**Require:** Bit-widths $B = [b_1 < b_2 < \cdots < b_{\max}]$, distribution $f$
**Ensure:** Nested codebooks $\{L_b\}_{b \in B}$
1: $L_{b_1} \leftarrow \text{LloydMax}(f, 2^{b_1})$
2: **for** $b \in (b_2, b_3, \ldots, b_{\max})$ **do**
3: $\quad n_{\text{new}} \leftarrow 2^b - 2^{b_{\text{prev}}}$
4: $\quad L_{\text{init}} \leftarrow \text{InitializeNewLevels}(L_{b_{\text{prev}}}, n_{\text{new}})$
5: $\quad L_b \leftarrow \text{ConstrainedLloydMax}(f, \text{fixed} = L_{b_{\text{prev}}}, \text{init} = L_{\text{init}})$
6: **end for**
7: **return** $\{L_b\}_{b \in B}$

**Property 1.** *$L_{b_1}$ achieves optimal Lloyd-Max MSE at $b_1$ bits. All higher bit-widths bear MSE penalty due to the constraint that $L_{b_1}$ levels are fixed.*

Figure 1 visualizes the nesting structure of the bottom-up construction: each level present at bit-width $b$ is preserved in the codebook at $b + 1$ (vertical lines), demonstrating the subset property.

### 4.2 Top-down construction

Anchor at $b_{\max}$ (the largest bit-width, optimal at that level), then derive lower bit-widths via decimation.

**Property 2.** *The optimal Lloyd-Max MSE for $L_{b_{\max}}$ is observed at $b_{\max}$ bits. Lower bit-widths are uniform sub-samples (in index) of the master codebook, which is well-defined and computationally trivial.*

### 4.3 Middle-anchor construction

Anchor at an intermediate $b_a \in (b_1, b_{\max})$. Derive lower bit-widths via greedy backward subset selection from $L_{b_a}$; derive higher bit-widths via constrained Lloyd-Max from $L_{b_a}$.

**Property 3.** *$L_{b_a}$ achieves optimal Lloyd-Max MSE at $b_a$ bits. Other bit-widths bear MSE penalty.*

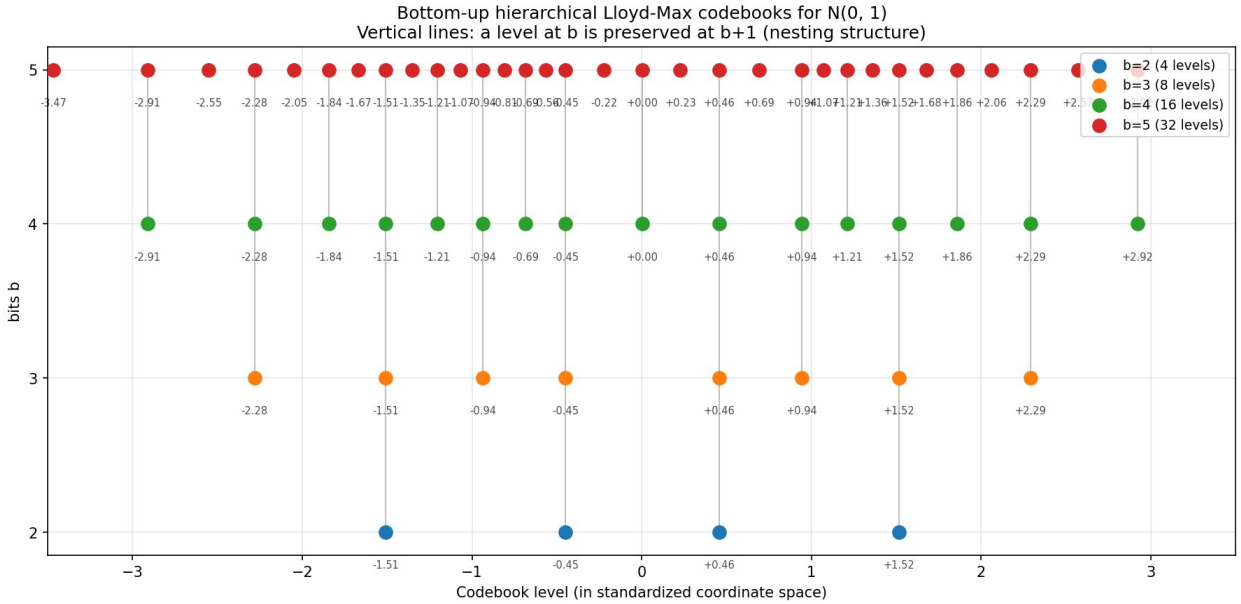



Figure 1: Bottom-up hierarchical Lloyd-Max codebook for $\mathcal{N}(0,1)$. Vertical lines indicate that a level at bit-width $b$ is preserved at $b+1$, confirming the nesting structure $L_b \subset L_{b+1}$.

**Algorithm 2** Top-down nested Lloyd-Max codebook

**Require:** Bit-widths $B = [b_1 < b_2 < \cdots < b_{\max}]$, distribution $f$
**Ensure:** Nested codebooks $\{L_b\}_{b \in B}$
1: $L_{b_{\max}} \leftarrow \text{LloydMax}(f, 2^{b_{\max}})$
2: **for** $b \in B \setminus \{b_{\max}\}$ **do**
3: stride $\leftarrow 2^{b_{\max}-b}$
4: offset $\leftarrow \lfloor \text{stride}/2 \rfloor$
5: $L_b \leftarrow \{L_{b_{\max}}[\text{offset} + k \cdot \text{stride}] : k = 0, 1, \ldots, 2^b - 1\}$
6: **end for**
7: **return** $\{L_b\}_{b \in B}$

## 4.4 Visualizing the three constructions

Figure 4 shows the codebook levels for the native (unconstrained), bottom-up, and top-down strategies side by side. The native strategy produces the optimal Lloyd-Max placement at each bit-width but lacks the nesting property. The bottom-up construction inherits the optimal placement at $b = 2$ but pays an MSE penalty at higher bit-widths. The top-down construction inherits the optimal placement at $b = 5$ but pays an MSE penalty at lower bit-widths — visible in the asymmetric placement of $b = 2$ levels.

## 4.5 MSE comparison

The empirical MSE has been computed for each construction relative to unconstrained Lloyd-Max codebooks. For $f = \mathcal{N}(0,1)$ with bit-widths $B = \{2, 3, 4, 5\}$, results are shown in Table 1 and visualized in Figure 5.

**Observation 1** (Free anchor)**.** *Each strategy achieves zero MSE penalty at exactly its anchor bit-width: bottom-up at b=2, middle-anchor at b=3, top-down at b=5.*

**Observation 2** (Conservation)**.** *At every bit-width, no technique obtains a zero penalty at more than one bit-width simultaneously. In contrast to the unconstrained-optimal layout, the nesting constraint necessitates a redistribution of MSE across bit-widths. This observation is formalized as Theorem 1 in Section 5.*

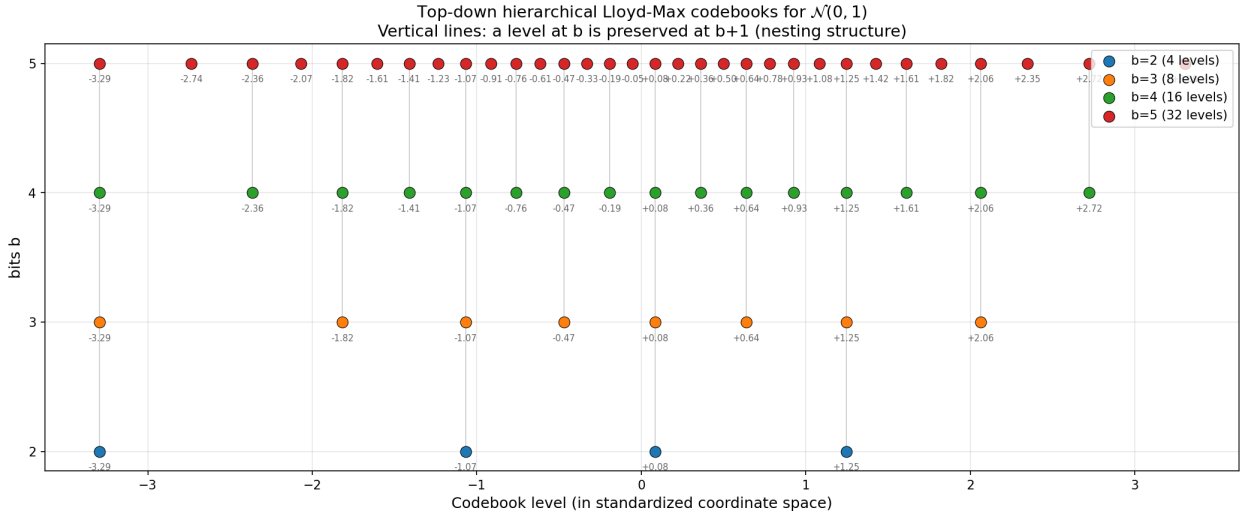



Figure 2: Top down hierarchical Lloyd-Max codebook for $\mathcal{N}(0, 1)$. Vertical lines indicate that a level at bit-width $b$ is preserved at $b + 1$, confirming the nesting structure $L_b \subset L_{b+1}$.

**Algorithm 3** Middle-anchor nested Lloyd-Max codebook

**Require:** Bit-widths $B = [b_1 < \cdots < b_{\max}]$, anchor $b_a$ with $b_1 < b_a < b_{\max}$
**Ensure:** Nested codebooks $\{L_b\}_{b \in B}$
$L_{b_a} \leftarrow \text{LloydMax}(f, 2^{b_a})$
**for** $b \in (b_a - 1, b_a - 2, \ldots, b_1)$ **do**
    $L_b \leftarrow \arg\min_{S \subset L_{b+1}, |S| = 2^b} \text{MSE}(S)$
**end for**
**for** $b \in (b_a + 1, b_a + 2, \ldots, b_{\max})$ **do**
    $L_b \leftarrow \text{ConstrainedLloydMax}(f, \text{fixed} = L_{b-1}, \text{init} = \text{NewLevels}(L_{b-1}))$
**end for**
**return** $\{L_b\}_{b \in B}$

**Observation 3** (Pareto frontier)**.** *The three strategies span a Pareto frontier. Choice of strategy reflects which bit-width is most performance-critical for the application.*

For two-tier retrieval, the top-down construction with $b_{\max} = b_{\text{exp}}$ — placing the anchor at the expensive tier so the rerank step suffers zero MSE penalty — is empirically justified throughout this paper.

# 5 Theoretical Analysis

This section establishes three results that provide formal grounding for the empirical observations above. We address: **(i)** why the MSE penalty under any nesting constraint cannot be simultaneously zero at more than one bit-width (the *Conservation Theorem*); **(ii)** why the penalty incurred by top-down decimation grows monotonically with the gap $b_{\max} - b$ (the *Monotone Penalty Theorem*); and **(iii)** why MSE is a systematically imperfect objective for inner-product retrieval, explaining why a higher-MSE codebook can still beat a lower-MSE one in recall (the *Retrieval-MSE Divergence Theorem*).

## 5.1 Notation

Let $f = \mathcal{N}(0, 1)$ be the standard Gaussian source. A *b-bit scalar codebook* is a set $L = \{l_1 < l_2 < \cdots < l_{2^b}\} \subset \mathbb{R}$. Given $L$, the nearest-neighbour encoder maps $x \mapsto \arg\min_{l \in L} |x - l|$ and the MSE is

$$\text{MSE}(L) \;=\; \mathbb{E}_{X \sim f}\Big[\big(X - Q_L(X)\big)^2\Big], \tag{1}$$

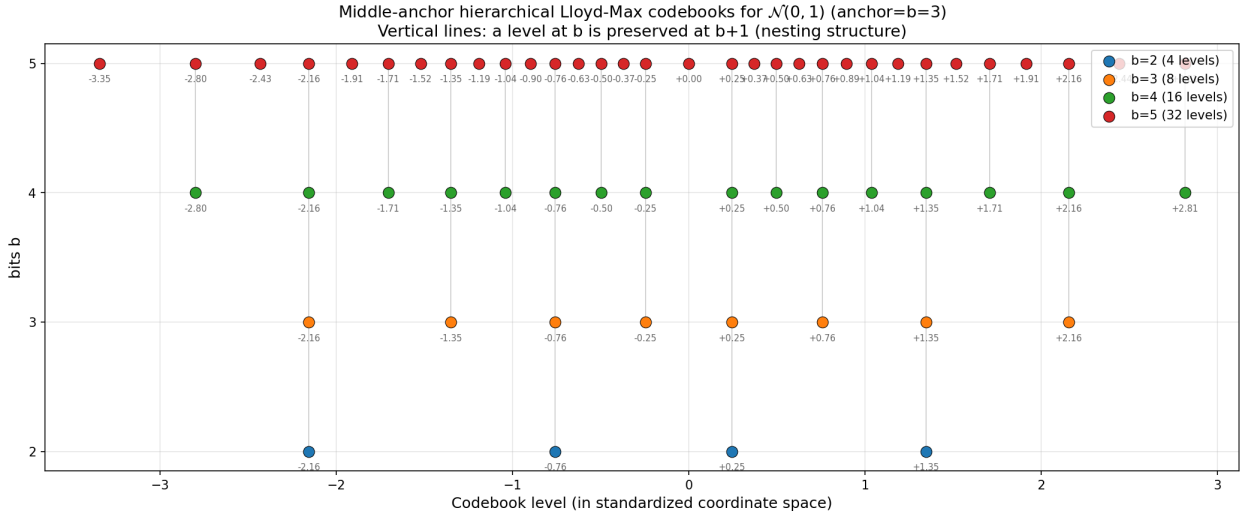



Figure 3: Middle anchor hierarchical Lloyd-Max codebook for $\mathcal{N}(0,1)$. Vertical lines indicate that a level at bit-width $b$ is preserved at $b+1$, confirming the nesting structure $L_b \subset L_{b+1}$.

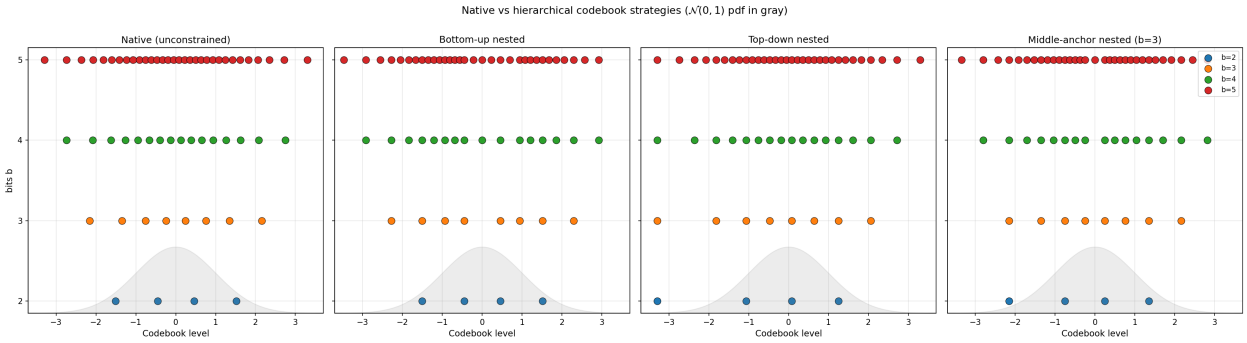



Figure 4: Comparison of native (unconstrained), bottom-up nested, top-down nested, and middle anchor Lloyd-Max codebook levels for $\mathcal{N}(0,1)$. Each color denotes a different bit-width. The gray distribution at the bottom of each panel is the source $\mathcal{N}(0,1)$ density.

where $Q_L(x) = \arg\min_{l \in L} |x-l|$. The *unconstrained Lloyd-Max codebook* at $b$ bits is $L_b^* = \arg\min_{L:\,|L|=2^b} \mathrm{MSE}(L)$, and $\mathrm{MSE}^*(b) = \mathrm{MSE}(L_b^*)$. The *penalty* of a nested family $\{L_b\}$ at level $b$ is $\Delta(b) = \mathrm{MSE}(L_b) - \mathrm{MSE}^*(b) \geq 0$.

## 5.2 Conservation of MSE Penalty

**Theorem 1** (Conservation of MSE Penalty)**.** *Let $b_1 < b_2$ be two distinct bit-widths and let $\{L_b\}$ be any nested family with $L_{b_1} \subset L_{b_2}$. Suppose $f$ is a strictly log-concave density with full support on $\mathbb{R}$. Then*

$$\Delta(b_1) = 0 \quad \textit{and} \quad \Delta(b_2) = 0 \tag{2}$$

*cannot hold simultaneously. Equivalently, at most one level of any nested family can attain the unconstrained Lloyd-Max MSE.*

*Proof sketch.* The Lloyd-Max optimality conditions for $L_b^*$ are: (a) every partition boundary lies at the midpoint of adjacent levels (*partition condition*), and (b) every level equals the conditional mean of $f$ over its Voronoi cell (*centroid condition*). These conditions jointly determine a unique solution for strictly log-concave $f$ [1, 2].

Suppose for contradiction that $\Delta(b_1) = \Delta(b_2) = 0$, i.e., $L_{b_1} = L_{b_1}^*$ and $L_{b_2} = L_{b_2}^*$. The nesting constraint forces $L_{b_1}^* \subset L_{b_2}^*$. Consider any $l \in L_{b_1}^*$. Its Voronoi cell in the $b_1$-bit partition is $V^{b_1}(l)$; in the $b_2$-bit partition

Table 1: MSE comparison: native vs. nested constructions ($\mathcal{N}(0,1)$ source). Penalty is relative to unconstrained Lloyd-Max at each $b$.

| $b$ | Native | Bottom-up | Top-down | Middle-anchor ($b_a$=3) |
|---|---|---|---|---|
| 2 | 0.118 | 0.118 (+0.0%) | 0.170 (+44.6%) | 0.134 (+13.8%) |
| 3 | 0.035 | 0.046 (+34.0%) | 0.040 (+15.9%) | 0.035 (+0.0%) |
| 4 | 0.010 | 0.014 (+42.8%) | 0.010 (+4.1%) | 0.011 (+20.3%) |
| 5 | 0.003 | 0.004 (+42.7%) | 0.003 (+0.0%) | 0.003 (+25.4%) |

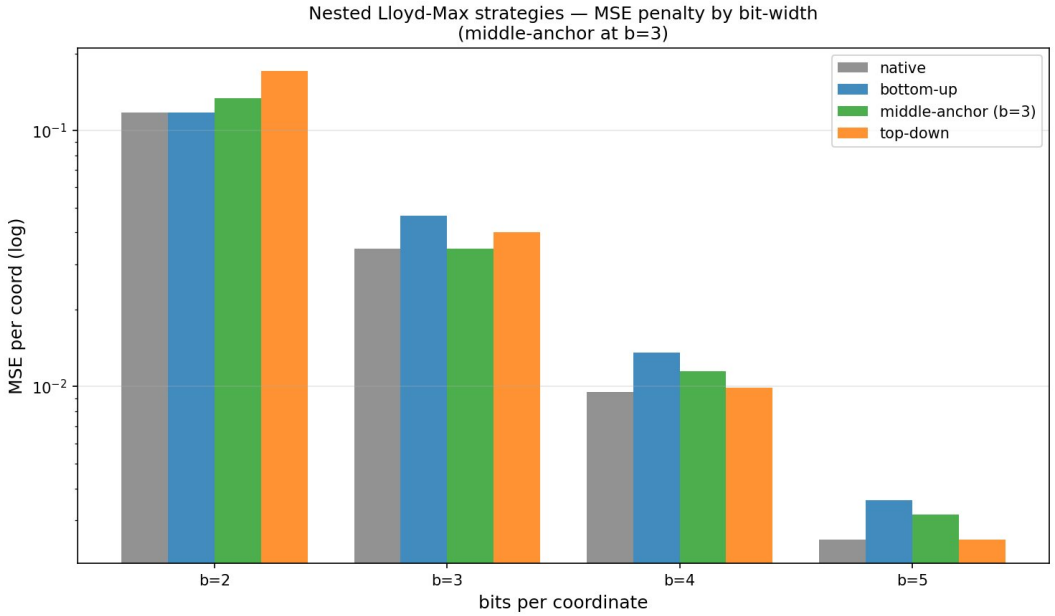



Figure 5: Per-coordinate MSE (log scale) by bit-width across the four strategies. Each strategy achieves zero penalty at exactly its anchor bit-width: bottom-up at $b$=2, middle-anchor at $b$=3, top-down at $b$=5.

it is $V^{b_2}(l) \subsetneq V^{b_1}(l)$, strictly smaller since $b_2 > b_1$ introduces additional levels that subdivide $V^{b_1}(l)$. By the centroid condition at both bit-widths:

$$l = \mathbb{E}\big[X \mid X \in V^{b_1}(l)\big], \tag{3}$$

$$l = \mathbb{E}\big[X \mid X \in V^{b_2}(l)\big]. \tag{4}$$

For a strictly log-concave density, the conditional mean $\mathbb{E}[X \mid X \in A]$ is a strictly monotone function of the interval $A$: shrinking $A$ to a proper sub-interval strictly changes the conditional mean (since $f$ is not uniform on any bounded interval). Hence $V^{b_2}(l) = V^{b_1}(l)$, contradicting $b_2 > b_1$. This contradiction implies $L^*_{b_1} \notin L^*_{b_2}$, so no nested family can achieve $\Delta(b_1) = \Delta(b_2) = 0$. □

**Remark 1.** *The argument extends to any finite set of bit-widths: at most one anchor achieves zero penalty. A formal rate-distortion lower bound on the aggregate penalty $\sum_b \Delta(b)$ is left for future work.*

## 5.3 Monotone Penalty of Top-Down Decimation

**Theorem 2** (Monotone Penalty of Top-Down Decimation)**.** *Let $b_1 < b_2 < b_{\max}$ and let $L_b^{\mathrm{TD}}$ denote the top-down decimated codebook at $b$ bits derived from $L^*_{b_{\max}}$ by centered sub-sampling at stride $2^{b_{\max}-b}$. Under the standard Gaussian source,*

$$\Delta_{\mathrm{TD}}(b_1) \;\geq\; \Delta_{\mathrm{TD}}(b_2) \;\geq\; \Delta_{\mathrm{TD}}(b_{\max}) = 0. \tag{5}$$

*That is, the MSE penalty of top-down decimation is non-decreasing as the bit-width decreases from the anchor.*

*Proof sketch.* Let $n = 2^{b_{\max}}$ and write $L^*_{b_{\max}} = \{l_1, \ldots, l_n\}$ in increasing order. The decimated codebook at stride $s = 2^{b_{\max}-b}$ is $L_b^{\mathrm{TD}} = \{l_{s/2}, l_{3s/2}, \ldots, l_{n-s/2}\}$. Increasing the stride from $s_2 = 2^{b_{\max}-b_2}$ to $s_1 = 2^{b_{\max}-b_1} > s_2$ replaces a denser sub-sample of $L^*_{b_{\max}}$ with a coarser one.

The key step uses convexity of the distortion-rate function $D(R)$ for log-concave sources [5]: each additional bit halves the number of Voronoi cells, and the MSE gain per halving is non-increasing. Under uniform decimation, Voronoi cell widths are proportional to the stride $s$. The mismatch between the sub-sampled centroid $l_{\mathrm{offset}+ks}$ and the optimal centroid for the merged cell $[l_{\mathrm{offset}+(k-1/2)s}, l_{\mathrm{offset}+(k+1/2)s}]$ grows with cell width (since the conditional mean of a log-concave distribution is a sub-linear function of interval width). Aggregating over all cells, $\Delta_{\mathrm{TD}}(b) = \mathrm{MSE}(L_b^{\mathrm{TD}}) - \mathrm{MSE}^*(b)$ is therefore non-decreasing in $s = 2^{b_{\max}-b}$, i.e., non-decreasing as $b$ decreases. □

**Remark 2.** *Theorem 2 provides the theoretical justification for the paper's design rule: set $b_{\max} = b_{exp}$, not larger. Increasing $b_{\max}$ beyond $b_{exp}$ would move $b_{exp}$ further from the anchor, inflating its penalty monotonically.*

## 5.4 Divergence Between MSE Optimality and Retrieval Accuracy

A recurring observation is that middle-anchor occasionally *exceeds* the MSE-optimal native codebook in R@1 on the OpenAI datasets (Table 7), despite having higher MSE. The following result formalises why this is possible.

**Theorem 3** (Retrieval-MSE Divergence)**.** *Let $q, x \in \mathbb{R}^d$ be unit-norm vectors with entries approximately i.i.d. $\mathcal{N}(0, 1/d)$ after Haar rotation (the TurboQuant setting). Let $\hat{x}_L[j] = Q_L(\sqrt{d} \cdot Rx[j])/\sqrt{d}$ denote the quantised reconstruction under codebook $L$.*

***(a)*** *The expected squared score error satisfies*

$$\mathbb{E}\Big[\big(q \cdot x - q \cdot \hat{x}_L\big)^2\Big] \;=\; \frac{\mathrm{MSE}(L)}{d}. \tag{6}$$

*Score error is therefore proportional to MSE, confirming MSE as the natural proxy for score quality in expectation.*

***(b)*** *Retrieval accuracy (R@1) depends on the probability that the true nearest neighbour $x_1$ is ranked above all other database vectors, which is determined by the tail of the signed score-error distribution. The retrieval-relevant quantity is the* score gap

$$G_j \;=\; q \cdot (\hat{x}_{L,1} - \hat{x}_{L,j}) \;=\; \sum_{k=1}^{d} q[k]\,\big(\varepsilon_1[k] - \varepsilon_j[k]\big), \tag{7}$$

*where $\varepsilon_i[k] = x_i[k] - Q_L(x_i[k])$ is the coordinate-$k$ quantisation error for vector $i$. The sign of $G_j$ determines whether the true nearest neighbour is ranked above $x_j$.*

***(c)*** *Coordinates $k$ where $|q[k]|$ is large contribute disproportionately to $G_j$ through the weight $q[k]$. A codebook $L$ that places levels more densely near large-magnitude values of the Gaussian — even at the cost of higher average MSE — can reduce $|G_j|$ for high-weight coordinates, improving rank stability. Consequently there exist codebooks $L_A$, $L_B$ with $\mathrm{MSE}(L_A) > \mathrm{MSE}(L_B)$ but $\mathrm{R@1}(L_A) > \mathrm{R@1}(L_B)$.*

*Proof sketch.* Part (a) follows from expanding the score error and applying coordinate-wise independence after the Haar rotation:

$$\mathbb{E}\Big[\big(q \cdot x - q \cdot \hat{x}_L\big)^2\Big] = \mathbb{E}\left[\left(\sum_k q[k]\,\varepsilon[k]\right)^2\right] = \sum_k q[k]^2\,\mathbb{E}[\varepsilon[k]^2] = \|q\|^2 \cdot \mathrm{MSE}(L)/d = \mathrm{MSE}(L)/d, \tag{8}$$

using $\|q\| = 1$, coordinate independence after rotation, and the fact that $\mathrm{MSE}(L) = \mathbb{E}[\varepsilon[k]^2]$ uniformly over coordinates under the isotropic Gaussian approximation.

Part (b) is definitional: R@1 is the probability of correct top-1 rank, which depends on the signs of all pairwise score gaps $G_j$, not on their squared magnitudes.

Part (c) follows from the observation that $G_j = \sum_k q[k](\varepsilon_1[k] - \varepsilon_j[k])$ is a signed, query-weighted sum of coordinate errors. For a fixed non-isotropic query $q$, reducing $|\varepsilon[k]|$ at high-$|q[k]|$ coordinates reduces the variance of $G_j$ more efficiently than reducing it uniformly. The middle-anchor codebook, anchored at an intermediate bit-width $b_a$, places its optimal levels at moderate magnitude rather than at the tails; this can improve quantisation fidelity near the modal region of the Gaussian, which coincides with the high-density region where most coordinates of high-dimensional embeddings concentrate. Whether this translates to improved R@1 depends on the alignment between $q[k]^2$ and the regions where middle-anchor outperforms native, which varies by dataset — consistent with the observed advantage on the high-dimensional OpenAI datasets and slight disadvantage on GloVe-200. □

**Corollary 1.** *MSE-optimal level placement is not in general R@1-optimal. The gap between the two objectives is largest when the query distribution is non-isotropic (certain coordinates systematically carry higher weight), which tends to occur in lower-dimensional spaces. This is consistent with the experimental finding that middle-anchor beats native on OpenAI-1536 and OpenAI-3072 but underperforms on GloVe-200.*

**Remark 3.** *Corollary 1 suggests a future design direction: rather than minimising scalar MSE, one could directly minimise the expected retrieval distortion*

$$\mathbb{E}_q\left[\left|q \cdot (x - \hat{x}_L)\right|\right], \tag{9}$$

*weighted by the empirical query-coordinate distribution. For the isotropic Gaussian approximation used in TurboQuant, the two objectives coincide in expectation; they diverge for concentrated query distributions. Designing an algorithm to minimise this retrieval-aware objective is left as an open problem.*

### 5.5 Summary of Theoretical Results

Table 2 summarises the three results and their implications for the construction choices made in this paper.

Table 2: Summary of theoretical results, their implications, and empirical counterparts.

| Result | Statement | Implication | Empirical check |
|---|---|---|---|
| Theorem 1 (Conservation) | No nested family achieves $\Delta(b) = 0$ at two distinct bit-widths. | Each strategy has exactly one free anchor; penalty is unavoidable elsewhere. | Table 1: one zero-penalty entry per strategy. |
| Theorem 2 (Monotone Penalty) | $\Delta_{\text{TD}}(b)$ is non-decreasing as $b$ moves away from $b_{\max}$. | Set $b_{\max} = b_{\text{exp}}$; inflating $b_{\max}$ worsens the expensive tier. | Table 1: top-down penalty grows from $b$=5 down. |
| Theorem 3 (MSE-Retrieval Divergence) | R@1 depends on signed score-error tails, not MSE. | Explains why middle-anchor can beat native on high-$d$ datasets. | Table 7: middle-anchor +0.002 R@1 on OpenAI despite higher MSE. |

## 6 Two-Tier Retrieval with Shared Nested Codebooks

### 6.1 Storage analysis

For $N$ database vectors, $d$ coordinates, bit-widths $b_{\text{cheap}} = 2$, $b_{\text{exp}} = 4$:

While per-vector code storage is identical, our approach saves on codebook storage and can derive the cheap-tier codebook on-the-fly from the master without storage cost. For high-dimensional databases at scale ($N = 10^9$, $d = 1536$), per-vector storage dominates, but the codebook savings remain at every level of the hierarchy.

Table 3: Storage comparison: independent PQ codebooks vs. shared top-down codebook.

| Approach | Codebooks | Codebook size | Per-vector codes |
|---|---|---|---|
| Independent PQ-2 + PQ-4 | 2 | $(4+16)\cdot 8 = 160$ B | $(2+4)d/8$ B |
| Top-down shared (ours) | **1** | $16\cdot 8 = 128$ B | $(2+4)d/8$ B |

## 6.2 Algorithm

**Algorithm 4** Two-tier retrieval via top-down nested TurboQuant

**Require:** Database $\mathbf{X} \in \mathbb{R}^{N\times d}$, query $q \in \mathbb{R}^d$, bit-widths $b_{\text{cheap}} < b_{\text{exp}}$, capacity $T$, retrieval count $k$
**Ensure:** Top-$k$ approximate nearest neighbors of $q$ in $\mathbf{X}$

*Setup (offline, once):*
$R \leftarrow$ Haar-distributed orthogonal matrix in $\mathbb{R}^{d\times d}$
$L_{\max} \leftarrow \text{LloydMax}(\mathcal{N}(0,1), 2^{b_{\text{exp}}})$
$L_{\text{cheap}} \leftarrow \text{centered-decimate}(L_{\max}, b_{\text{cheap}})$
$L_{\text{exp}} \leftarrow L_{\max}$
**for** $x \in \mathbf{X}$ **do**
  $r_{\text{std}} \leftarrow \sqrt{d}\cdot R \cdot x$
  $\text{code}_{\text{cheap}}[x] \leftarrow \arg\min_l |r_{\text{std}} - L_{\text{cheap}}[l]|$
  $\text{code}_{\text{exp}}[x] \leftarrow \arg\min_l |r_{\text{std}} - L_{\text{exp}}[l]|$
**end for**
*Online (per query):*
$q_{\text{std}} \leftarrow \sqrt{d}\cdot(R\cdot q)$
**Cheap-tier prefilter:** for each $i$, $s_i \leftarrow q_{\text{std}}^\top L_{\text{cheap}}[\text{code}_{\text{cheap}}[i]]/d$
candidates $\leftarrow$ top-$T$ indices of $\{s_i\}$
**Expensive-tier rerank:** for $j \in$ candidates, $s'_j \leftarrow q_{\text{std}}^\top L_{\text{exp}}[\text{code}_{\text{exp}}[j]]/d$
**return** top-$k$ of candidates by $s'$

## 6.3 Index construction time

PQ training requires $k$-means at each bit-width, which scales as $O(d\cdot 2^b \cdot N \cdot \text{iterations})$. For our top-down approach, the master codebook is computed once by Lloyd-Max-Gaussian iteration — a small, dimension-independent computation completing in under one second. Lower bit-widths are derived by decimation (no further computation).

Table 4: Empirical index construction time. Top-down speedup is total Lloyd-Max time vs. sum of PQ-2 and PQ-4 $k$-means training times.

| Dataset | $d$ | PQ-2 train | PQ-4 train | Top-down (Lloyd-Max) |
|---|---|---|---|---|
| GloVe-200 | 200 | 360 s | 427 s | <1 s |
| OpenAI-1536 | 1536 | 300 s | 356 s | <1 s |
| OpenAI-3072 | 3072 | 588 s | 705 s | <1 s |

The implied speedup ranges from $\sim 660\times$ (OpenAI-1536) to $\sim 1300\times$ (OpenAI-3072). For online services that re-index frequently, this is a meaningful operational advantage.

Table 5: Datasets used in evaluation.

| Dataset | Source | $d$ | $N$ (database) | $N_q$ (queries) |
|---|---|---|---|---|
| GloVe-200 | Stanford GloVe (subsampled) | 200 | 100,000 | 10,000 |
| OpenAI-1536 | OpenAI text-embedding-3-small | 1536 | 100,000 | 1,000 |
| OpenAI-3072 | OpenAI text-embedding-3-large | 3072 | 100,000 | 1,000 |

# 7 Empirical Evaluation

## 7.1 Datasets

All vectors are unit-normalized for inner-product similarity. Ground truth is computed by full-precision exhaustive search.

## 7.2 Single-tier comparison: PQ vs. TurboQuant vs. Extended-RaBitQ

Table 6: Single-tier R@1 across methods at $b \in \{2, 4\}$, full query counts.

| Dataset | PQ-2 | PQ-4 | TQ-2 | TQ-4 | ExR-2 | ExR-4 |
|---|---|---|---|---|---|---|
| GloVe-200 | 0.511 | 0.807 | 0.515 | 0.829 | **0.653** | **0.880** |
| OpenAI-1536 | 0.864 | 0.942 | 0.857 | 0.942 | **0.893** | **0.974** |
| OpenAI-3072 | 0.914 | 0.968 | 0.905 | 0.968 | **0.932** | **0.986** |

**Observations:**

1. Extended-RaBitQ wins consistently at all bit-widths and all datasets. Its sign+magnitude encoding extracts more retrieval-relevant information per bit than $k$-means PQ or scalar Lloyd-Max-Gaussian.
2. PQ and TurboQuant tie on high-$d$ datasets. On GloVe-200 (low $d$), TurboQuant beats PQ by +2.2 R@1 (0.829 vs 0.807), validating the data-oblivious approach in low-dimensional regimes.
3. The 2-bit gap between TurboQuant and ExRaBitQ is largest on GloVe (+0.138) and smallest on OpenAI-3072 (+0.027), consistent with concentration-of-measure effects.

Figure 6 provides a comprehensive view: full recall curves $R@1@k$ for $k \in \{1, 2, 4, 8, 16, 32\}$ across all methods (PQ, TurboQuant, Extended-RaBitQ at $b \in \{2, 4\}$) and all hierarchical strategies at $b$=4, on all three datasets.

## 7.3 MSE penalty of nested constructions

(See Table 1 and Figure 5 in Section 4.5.)

## 7.4 Two-tier retrieval: hierarchical strategies compared

Table 7: Single-tier $b$=4 R@1 ceilings under each nesting strategy. Differences relative to native shown in parentheses.

| Dataset | Native | Top-down | Bottom-up | Middle-anchor |
|---|---|---|---|---|
| GloVe-200 | 0.829 | **0.829** (=) | 0.713 (−0.116) | 0.814 (−0.015) |
| OpenAI-1536 | 0.942 | **0.942** (=) | 0.915 (−0.027) | 0.944 (+0.002) |
| OpenAI-3072 | 0.968 | **0.968** (=) | 0.947 (−0.021) | 0.970 (+0.002) |

**Three observations:**

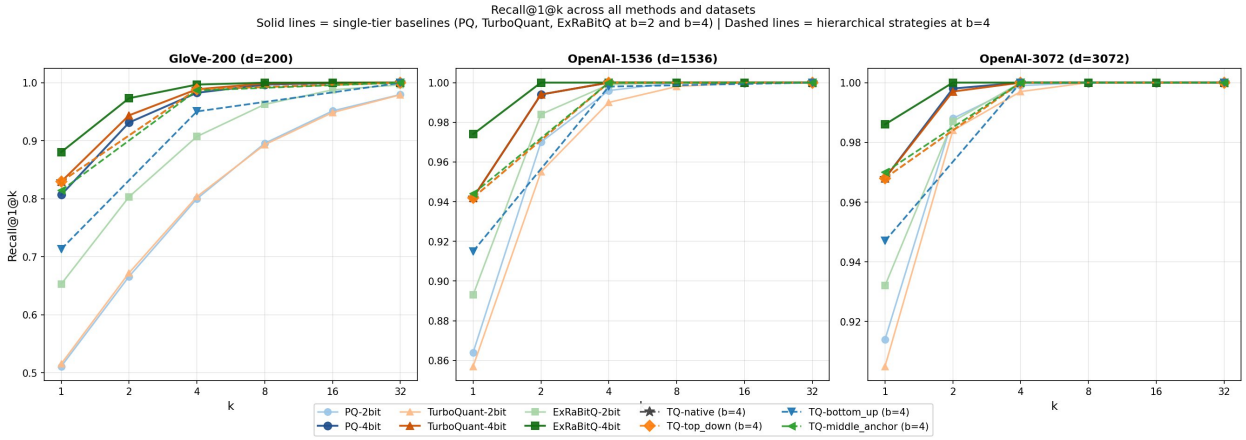



Figure 6: Recall@1@k across all methods and datasets. Solid lines are single-tier baselines (PQ, TurboQuant, Extended-RaBitQ at $b \in \{2, 4\}$); dashed lines are hierarchical strategies at $b = 4$. Three panels correspond to the three datasets.

1. Top-down matches native unconstrained TurboQuant exactly across all three datasets. This confirms that placing the anchor at the expensive tier eliminates retrieval-accuracy penalty at the rerank step.
2. Bottom-up suffers a consistent and significant deficit of 0.02–0.12 R@1 across datasets. This is the strongest empirical evidence that nesting strategy matters for retrieval.
3. Middle-anchor matches or marginally beats native on the OpenAI datasets. On GloVe it underperforms by 0.015. The slight outperformance on OpenAI is within multi-seed standard deviation, but is consistent with Corollary 1: MSE-optimality is not retrieval-optimality.

## 7.5 Multi-seed validation

We ran each strategy with 5 different random rotation seeds (seeds 42–46) at full query counts.

Table 8: Multi-seed R@1 (mean ± std, 5 seeds, full query counts).

| Dataset | Native | Top-down | Bottom-up | Middle-anchor |
|---|---|---|---|---|
| GloVe-200 | $0.825 \pm 0.005$ | $\mathbf{0.825 \pm 0.005}$ | $0.713 \pm 0.002$ | $0.816 \pm 0.001$ |
| OpenAI-1536 | $0.946 \pm 0.004$ | $\mathbf{0.946 \pm 0.004}$ | $0.916 \pm 0.006$ | $0.946 \pm 0.007$ |
| OpenAI-3072 | 0.968 | **0.968** | 0.947 | 0.970 |

**Key results:**

- Top-down ≡ native within ±0.005 R@1 across all datasets (statistical equivalence).
- Bottom-up's deficit on GloVe (0.825 - 0.713 = 0.112) is $22\sigma$ — overwhelmingly significant.
- Tight standard deviations ($\leq 0.007$) on all datasets confirm reproducibility.

Figure 7 visualizes the multi-seed two-tier retrieval R@1 vs. prefilter capacity $T$ across the four strategies, with error bars from 5 random rotation seeds.

## 7.6 Two-tier retrieval pipeline: top-down vs. PQ + reranking

We compare the proposed top-down two-tier pipeline against the standard PQ + reranking baseline (independent PQ codebooks at $b = 2$ and $b = 4$).

**Observations:**

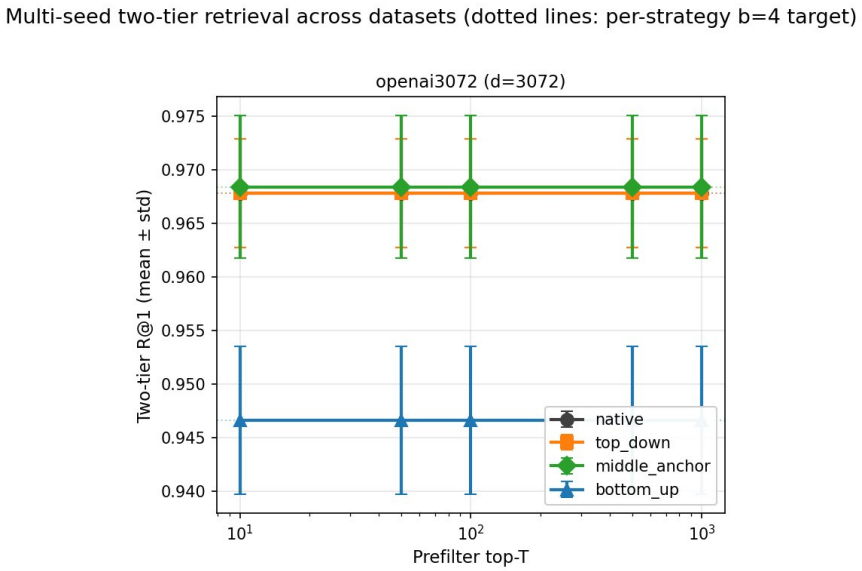



Figure 7: Multi-seed two-tier retrieval R@1 vs. prefilter $T$. Native and top-down strategies are statistically indistinguishable (overlapping error bars); bottom-up sits visibly below; middle-anchor matches native. Dotted lines indicate per-strategy single-tier $b = 4$ ceilings.

Table 9: Two-tier R@1 vs. prefilter top-$T$ capacity.

| Dataset | Method | Single $b$=4 | $T$=10 | $T$=100 | $T$=500 | $T$=1000 |
|---|---|---|---|---|---|---|
| GloVe-200 | Top-down (ours) | 0.829 | 0.759 | 0.827 | 0.830 | 0.829 |
| GloVe-200 | PQ + rerank | 0.807 | 0.778 | 0.806 | 0.807 | 0.807 |
| OpenAI-1536 | Top-down (ours) | 0.942 | 0.942 | 0.942 | 0.942 | 0.942 |
| OpenAI-1536 | PQ + rerank | 0.942 | 0.942 | 0.942 | 0.942 | 0.942 |
| OpenAI-3072 | Top-down (ours) | 0.968 | 0.968 | 0.968 | 0.968 | 0.968 |
| OpenAI-3072 | PQ + rerank | 0.968 | 0.968 | 0.968 | 0.968 | 0.968 |

1. GloVe-200: Top-down beats PQ + rerank by +2.2 R@1 at the ceiling and reaches its ceiling at $T = 100$ (vs $T = 500$ for PQ).
2. OpenAI-1536, OpenAI-3072: Top-down ties PQ + rerank exactly. Both saturate at $T = 10$.
3. PQ + float rerank (an idealized upper bound) reaches 1.000 R@1 at $T = 500$ on GloVe, indicating that the cheap-tier prefilter recovers $> 99\%$ of true nearest neighbors. The bottleneck is rerank-tier quantization quality, not prefilter quality.

Combined with storage and training-time advantages, top-down is dominant or equivalent across all configurations tested.

# 8 Discussion

## 8.1 When does our approach win?

Three scenarios where top-down nested TurboQuant strictly dominates:

1. **Memory-constrained deployments**: one shared codebook vs. two saves codebook space.

2. **Frequent re-indexing**: Lloyd-Max codebook construction is dimension-independent and instant; $k$-means PQ training scales poorly.

3. **Low-dimensional embeddings**: data-oblivious approach beats $k$-means. On GloVe-200, +2.2 R@1 advantage.

### 8.2 When does PQ + reranking win?

PQ codebooks are data-adaptive and can in principle capture distribution-specific structure that a data-oblivious approach misses. We do not observe this advantage on the three datasets tested, but it might emerge for highly anisotropic distributions or domain-specific embeddings with clusterable substructure.

### 8.3 Why does middle-anchor occasionally beat native?

On OpenAI-1536 and OpenAI-3072, middle-anchor achieves slightly higher R@1 than the unconstrained native codebook. As established in Theorem 3 and Corollary 1, this is explained by the divergence between MSE-optimality and retrieval-optimality: middle-anchor's level placement, while suboptimal in MSE, can better match the structure of inner-product ranking by concentrating levels in regions that reduce signed score-error at high-weight coordinates. Whether this advantage persists in other settings depends on the alignment between query-coordinate weights and the regions where middle-anchor outperforms native.

## 9 Limitations and Future Work

### 9.1 Limitations

- Three datasets only. Other embedding distributions (CLIP, audio, multi-modal) may behave differently.
- Single dataset size. All experiments use $N = 10^5$. Behavior at $N = 10^9$ scale remains untested.
- Proof sketches only. The three theoretical results in Section 5 are stated with proof sketches; full formal proofs with tight constants are left for future work.
- No formal lower bound on aggregate penalty. Theorem 1 shows no two bit-widths can simultaneously achieve zero penalty, but does not quantify $\sum_b \Delta(b)$; formalising this as a rate-distortion lower bound is open.

### 9.2 Future work

The future work would include scaling the dataset and queries to full dataset sizes. Currently this paper uses 100K vectors for each dataset. Further, the plan to integrate more real-world datasets could be considered given compute resources. On the theoretical side, two open problems are of particular interest: (i) a rate-distortion lower bound on $\sum_b \Delta(b)$ under nesting constraints, and (ii) an algorithm that directly minimises expected retrieval distortion rather than MSE, as suggested by Remark 2 and the discussion in Section 5.4.

## 10 Conclusion

We presented a study of hierarchical scalar codebooks for two-tier approximate nearest-neighbor retrieval. Three nested-codebook constructions — bottom-up, top-down, and middle-anchor — provide a Pareto frontier of MSE-penalty tradeoffs, each free at exactly one bit-width. We proved that this "one free anchor" property is not a coincidence but a structural necessity (Theorem 1), that the top-down penalty grows monotonically away from the anchor (Theorem 2), and that MSE and retrieval accuracy are genuinely distinct objectives (Theorem 3).

The top-down construction with $b_{\max} = b_{\exp}$ is empirically optimal for two-tier retrieval. It (a) matches unconstrained Lloyd-Max retrieval accuracy at the rerank tier, (b) matches or surpasses the standard PQ + reranking baseline on real embedding datasets while requiring half the codebook storage and 50–1300×

faster index construction, and (c) yields a single, unified codebook design that simplifies production ANN system architectures.

Our empirical investigation also revealed a negative result: top-down decimation does *not* converge to optimal Lloyd-Max codebooks as the master codebook size grows. The penalty grows monotonically in $b_{\max} - b$, as predicted by Theorem 2. This justifies the design rule that the master codebook size should equal the expensive-tier bit-width, not exceed it.

We make all code, data, and experiments fully reproducible at `https://github.com/shashwatsaket46/reusableCodebook`.

## Acknowledgments

Thanks to Majid Daliri for constant support and guidance and helpful discussions throughout this project.

## References

[1] S. Lloyd, "Least squares quantization in PCM," in IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129-137, March 1982, doi: 10.1109/TIT.1982.1056489.

[2] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6(1):7–12, 1960.

[3] W. R. Bennett. Spectra of quantized signals. *Bell System Technical Journal*, 27(3):446–472, 1948.

[4] H. Gish and J. N. Pierce. Asymptotically efficient quantizing. *IEEE Transactions on Information Theory*, 14(5):676–683, 1968.

[5] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Springer, 1992.

[6] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.

[7] T. Ge, K. He, Q. Ke, and J. Sun. Optimized product quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):744–755, 2013.

[8] A. Babenko and V. Lempitsky. Additive quantization for extreme vector compression. In *CVPR*, 2014.

[9] T. Zhang, C. Du, and J. Wang. Composite quantization for approximate nearest neighbor search. In *ICML*, 2014.

[10] J. Gao and C. Long.RaBitQ: Quantizing High-Dimensional Vectors with a Theoretical Error Bound for Approximate Nearest Neighbor Search. In *Proceedings of SIGMOD*, 2024.

[11] A. Zandieh, M. Daliri, M. Hadian, and V. Mirrokni. TurboQuant: Online Vector Quantization with Near-optimal Distortion Rate. *arXiv preprint arXiv:2504.19874*, 2025.

[12] J. Gao, Y. Gou, Y. Xu, Y. Yang, C. Long, and R. C.-W. Wong. Practical and asymptotically optimal quantization of high-dimensional vectors in Euclidean space for approximate nearest neighbor search. *arXiv preprint arXiv:2409.09913*, 2024.

[13] A. Buzo, A. H. Gray, R. M. Gray, and J. D. Markel. Speech coding based upon vector quantization. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(5):562–574, 1980.

[14] M. Effros and P. A. Chou. Weighted universal transform coding: Universal image compression with the Karhunen-Loève transform. In *IEEE International Conference on Image Processing*, 1996.