Gray-Tunneled Hashing: Elementary Landscapes, Planted Structure, and a Research Program for Binary Vector Search

Draft white paper for internal research

Carlos R. B. Azevedo

November 5, 2025

Abstract

Binary embeddings and Hamming-distance indexes are central to large-scale approximate nearest neighbor (ANN) search and in-memory vector databases. Yet the way binary codes are assigned to continuous embeddings is typically treated as an afterthought of quantization, rather than a primary object of optimization. We study the assignment problem explicitly. We model the code space as a Hamming hypercube and cast the assignment of embeddings to hypercube vertices as a Quadratic Assignment Problem (QAP). Under a natural 2-swap neighborhood on permutations, we show that the objective defines an elementary landscape in the sense of Whitley, and we derive the associated eigenvalue and drift equation.

Building on this structure, we propose a planted random model for "good" instances—where a pseudo-Gray assignment is hidden and embeddings are noisy observations—and formulate a notion of statistical tunneling via block moves: operators that reoptimize assignments on small substructures (clusters or subcubes) to escape poor local minima in typical instances. We argue that this framework underpins a practical algorithm family, Gray-Tunneled Hashing, which aims to improve binary ANN quality at fixed memory and compute budgets. We outline a research and experimental program to validate the theory and to deploy such algorithms in real vector database settings.

1 Introduction

Binary vector search is deeply intertwined with the rise of vector databases and large embedding models. In many production systems, continuous embeddings (e.g., from transformers) are stored only approximately, binary codes and Hamming distance are used as primary or pre-filter similarity measures, and libraries such as FAISS or modern vector stores provide specialized support for binary indexes.

In practice, however, the design of binary codes is often simplistic: direct sign-thresholding or random projections to bits; local k-means or product quantization whose indices happen to be encoded in binary; and very limited control over how Hamming distance relates to semantic similarity. Classical work on pseudo-Gray coding in vector quantization showed that the mapping from indices to binary labels matters: if neighboring codevectors are mapped to Hamming-nearby labels, a single bit flip or small channel error will result in limited distortion. That line of work mainly targeted signal coding and small codebooks.

We revisit these ideas in the context of modern embeddings and large-scale ANN, and we introduce a structural framework:

- We model the assignment of embeddings to hypercube vertices as a QAP whose objective encourages Hamming-1 neighbors to be semantically close.
- Under a 2-swap neighborhood on permutations, this objective defines an elementary landscape; we derive the associated spectral constant.
- We adopt a planted model where there is an underlying pseudo-Gray assignment aligning code geometry and embedding geometry, plus subgaussian noise, and we conjecture strong typical-case properties.
- We propose statistical tunneling via block moves—reoptimizing assignments on small blocks—to escape poor local minima and approach the planted configuration.
- On the algorithmic side, we outline Gray-Tunneled Hashing, a family of methods that explicitly optimize code assignment and plug into existing binary ANN infrastructure.

The goal of this white paper is to lay out the theory and the research program needed to go from this conceptual framework to a publishable theory and practical implementations.

2 Problem Formulation

2.1 Code space and embeddings

Let $X = \{x_1, \ldots, x_M\} \subset \mathbb{R}^d$ denote embeddings produced by some upstream model. We aim to assign binary codes $b_i \in \{0,1\}^n$ to each x_i such that:

- Hamming distance between codes correlates with semantic distance between embeddings;
- In particular, Hamming-1 neighbors correspond to small semantic distortions.

For the core theory we study the full hypercube regime $M = N = 2^n$, so that all hypercube vertices are used. We can relabel embeddings as $w_1, \ldots, w_N$. The hypercube vertices are $V = \{0,1\}^n$, which we identify with integers $1, \ldots, N$ when convenient. An assignment is a permutation $\pi \in S_N$ mapping vertices to embeddings: vertex u hosts embedding $w_{\pi(u)}$.

2.2 Hypercube graph and cost structure

Define the location graph as the n-dimensional hypercube Q_n:

- Vertices: $V = \{0,1\}^n$;
- Edges: $E = \{(u,v) : \|u - v\|_H = 1\}$, where $\|\cdot\|_H$ is Hamming distance.

Let the semantic cost matrix be

$$d_{ij} = \|w_i - w_j\|^2,$$

or more generally any nonnegative symmetric matrix reflecting semantic dissimilarity. Given an assignment π, we define the QAP objective

$$f(\pi) = \sum_{(u,v)\in E} d_{\pi(u)\,\pi(v)}. \tag{1}$$

This objective is small if embeddings placed at Hamming-1 neighbors are close. The optimization problem is

$$\min_{\pi \in S_N} f(\pi). \tag{2}$$

This is a QAP with: locations given by vertices in Q_n; facilities given by embedding indices $\{1, \ldots, N\}$; flow given by unit weight on hypercube edges; and distances given by d_{ij}. In general QAP is NP-hard, and we do not expect to solve worst-case instances exactly. Instead, we aim for typical-case guarantees under structural assumptions and for designing effective local-search algorithms with tunable neighborhoods.

3 Assumptions and Planted Model

To make rigorous statements about landscapes and local minima, we need a structural model of "typical" instances where good solutions exist and are not adversarially hidden.

3.1 Semantic metric and hypercube locality

We assume the matrix d_{ij} derives from a meaningful embedding metric: the embedding space is $\mathbb{R}^d$ with a norm such as ℓ_2 or cosine-based distances; and nearby embeddings correspond to semantically similar items. We also fix the hypercube structure Q_n as the code space: Hamming distance is the operational metric used by the ANN index. The goal is to align these two geometries as much as possible via the assignment π.

3.2 Ideal pseudo-Gray configuration

We posit the existence of an ideal pseudo-Gray configuration: a map $\phi : V \to \mathbb{R}^d$ such that

- if $\|u - v\|_H = 1$ then $\|\phi(u) - \phi(v)\|$ is small;
- if $\|u - v\|_H \geq 2$, the distances are typically larger, with a positive margin.

Informally, ϕ is a smooth embedding of the hypercube where edges of Hamming distance correspond to local moves in semantic space with controlled distortion.

In practice, ϕ could be:

- a low-dimensional embedding of Q_n with controlled edge lengths;
- the centroids of a VQ or PQ codebook designed to reflect clustering structure;
- the asymptotic mean of embeddings assigned to each code in a trained model.

3.3 Planted structure and noise

We adopt a planted model for the true embeddings. There exists a planted permutation π such that

$$w_{\pi^{(u)}} = \phi(u) + \xi_u, \qquad u \in V, \tag{3}$$

where ξ_u are independent noise vectors. We assume ξ_u are subgaussian with variance proxy σ^2, i.e., for all unit vectors a,

$$\mathbb{E}\big[\exp(t\langle a, \xi_u \rangle)\big] \leq \exp\left(\frac{\sigma^2 t^2}{2}\right). \tag{4}$$

We also impose margin conditions on ϕ: there exist constants $0 < \delta_1 < \delta_2$ such that

$$\|u - v\|_H = 1 \implies \|\phi(u) - \phi(v)\|^2 \le \delta_1, \tag{5}$$

$$\|u - v\|_H \ge 2 \implies \|\phi(u) - \phi(v)\|^2 \ge \delta_2. \tag{6}$$

We assume noise is small relative to margins,

$$\sigma^2 \ll \delta_2 - \delta_1. \tag{7}$$

Under such assumptions, we expect π to be (almost) globally optimal for the QAP, and assignments that disagree with π on a nontrivial fraction of vertices to incur a significant cost penalty.

Our theoretical program is to make such statements precise and to quantify cost gaps and local-minimum structure under this planted model.

4 Elementary Landscape under 2-Swap Moves

We now turn to the combinatorial landscape of $f(\pi)$ under local moves. We focus first on the simplest neighborhood: 2-swaps, i.e., transpositions of two vertices.

4.1 2-swap neighborhood and adjacency operator

The search space is S_N. For $\pi \in S_N$, define the neighborhood

$$N(\pi) = \{\pi' : \pi' = \pi \circ (u\, v),\ \text{for some } 1 \le u < v \le N\}, \tag{8}$$

where $(u\, v)$ denotes the transposition of vertices u and v. For each π, the number of neighbors is

$$d = |N(\pi)| = \binom{N}{2} = \frac{N(N-1)}{2}. \tag{9}$$

Define the neighbor-averaging operator L acting on functions $g : S_N \to \mathbb{R}$ by

$$(Lg)(\pi) = \frac{1}{d} \sum_{\pi' \in N(\pi)} g(\pi'). \tag{10}$$

We are interested in Lf, where f is our QAP objective.

4.2 Global mean of the objective

Let

$$\bar{f} = \mathbb{E}_{\pi \sim \mathrm{Unif}(S_N)}[f(\pi)]. \tag{11}$$

By symmetry of π, each unordered pair of embeddings $\{i, j\}$ is equally likely to land on any unordered pair of vertices $\{u, v\}$, and each edge receives a random pair drawn without replacement.

Let $E = |E(Q_n)|$ be the number of edges in the hypercube. For n dimensions, $E = \frac{N \log_2 N}{2}$. Let

$$S = \sum_{1 \le i < j \le N} d_{ij}. \tag{12}$$

Then

$$\bar{f} = \frac{E}{\binom{N}{2}} S = \frac{\frac{N \log_2 N}{2}}{\frac{N(N-1)}{2}} S = \frac{\log_2 N}{N-1} S, \tag{13}$$

independent of π.

4.3 Elementary landscape definition

A function $f : S_N \to \mathbb{R}$ is an elementary landscape (under a given neighborhood) if its neighbor average is an affine function of f:

$$(Lf)(\pi) = f(\pi) + \lambda(\bar{f} - f(\pi)) = (1-\lambda)f(\pi) + \lambda\bar{f}, \tag{14}$$

for some constant λ that does not depend on π.

In this case, $f - \bar{f}$ is an eigenfunction of the operator L with eigenvalue $1 - \lambda$:

$$L(f - \bar{f}) = (1-\lambda)(f - \bar{f}). \tag{15}$$

The constant λ can be interpreted as the relaxation rate: how fast repeated random neighbor steps contract deviations from the mean.

4.4 Main elementary landscape result

Theorem 1 (Elementary landscape for hypercube QAP under 2-swap). Let $f(\pi) = \sum_{(u,v)\in E(Q_n)} d_{\pi(u)\,\pi(v)}$ be defined as above. Under the 2-swap neighborhood and neighbor operator L, f is an elementary landscape:

$$(Lf)(\pi) = f(\pi) + \frac{4}{N}(\bar{f} - f(\pi)). \tag{16}$$

Equivalently,

$$\mathbb{E}[f(\pi') \mid \pi] = \left(1 - \frac{4}{N}\right) f(\pi) + \frac{4}{N}\,\bar{f}, \tag{17}$$

where π' is a uniformly random 2-swap neighbor of π. Thus, $f - \bar{f}$ is an eigenfunction of L with eigenvalue $1 - \frac{4}{N}$.

4.5 Proof sketch

We sketch a combinatorial argument; full details can be formalized via the representation theory of S_N, but this is not necessary to grasp the key idea.

1. **Decomposition by edges.** Write $f(\pi) = \sum_{e\in E} g_e(\pi)$, where $g_e(\pi) = d_{\pi(u_e)\,\pi(v_e)}$ for edge $e = (u_e, v_e)$.

2. **Effect of a transposition.** Consider a fixed edge $e = (u, v)$. For a given transposition $(a\,b)$, $g_e(\pi \circ (a\,b))$ equals $g_e(\pi)$ if neither u nor v is a or b, and some other value involving swapped embeddings otherwise. For a uniformly random transposition, the probability that the edge endpoints are unaffected is $\frac{\binom{N-2}{2}}{\binom{N}{2}}$, and the probability that exactly one endpoint is in $\{a, b\}$ or both are is easy to compute. The key point is that the expected contribution from affected transpositions aggregates to a simple affine function of the global average.

3. **Symmetry.** By symmetry of embeddings and vertices, the behavior for each edge is identical in expectation. Therefore, the neighbor-averaged change in g_e depends only on $g_e(\pi) - \bar{g}_e$, where $\bar{g}_e$ is the average over permutations, and the proportionality constant is the same across edges.

4. **Computation of λ.** A detailed counting argument shows that the eigenvalue corresponds to the probability that a given edge is "touched" by a random transposition, multiplied by a factor that accounts for the re-randomization effect. This calculation yields $\lambda = 4/N$.

A sanity check with small N and random distances confirms that for $N = 4$, $\lambda = 1$ (neighbor-averaged f is constant $\bar{f}$), and for $N = 8$, $\lambda = 1/2$ (deviations are halved at each step).

4.6 Implications

The elementary landscape structure implies:

- Random walks (or randomized local search) under 2-swap quickly mix around the global average $\bar{f}$.
- This is a global statement; it does not preclude the presence of many local minima far from π.
- The constant $\lambda = 4/N$ becomes small as N grows, indicating that a single 2-swap step has limited pull toward the average; large-scale rearrangements require many steps or more powerful moves.

This motivates the need for richer operators—block moves—if we want to escape poor local minima and exploit planted structure.

5 Block Moves and Statistical Tunneling

We now formalize the notion of tunneling via block moves: transitions that reconfigure the assignment on a small subset of vertices in a way that can jump across barriers in the local landscape.

5.1 Block neighborhoods

Let $B \subset V$ with $|B| = k$. Given an assignment π, a block move on B proceeds as follows:

1. Let I_B be the set of embeddings currently assigned to vertices in B: $I_B = \{\pi(u) : u \in B\}$.
2. Consider the restricted QAP:
$$f_B(\sigma) = \sum_{(u,v)\in E_B} d_{\sigma(u)\,\sigma(v)} + \sum_{\substack{u\in B,\, v\in V\setminus B \\ (u,v)\in E}} d_{\sigma(u)\,\pi(v)}, \tag{18}$$
where E_B is the set of edges with both endpoints in B, and σ ranges over permutations assigning the same set I_B of embeddings to vertices in B.
3. Solve (or approximately solve) the small QAP to find σ with minimal $f_B(\sigma)$.
4. Replace $\pi(u)$ by $\sigma(u)$ for all $u \in B$ if this reduces the global objective $f(\pi)$.

We then define a block neighborhood of π consisting of all assignments reachable via such a block reoptimization for some subset B of allowed shapes (e.g., clusters or small subcubes).

5.2 Choice of blocks

In practice, we do not consider all subsets B; we restrict to:

- **Cluster-based blocks.** Partition embeddings into clusters (e.g., via k-means in embedding space), and associate each cluster with a block. Blocks gather vertices assigned to embeddings in the same cluster.
- **Geometric subcubes.** Choose subsets of vertices that form small subcubes of dimension m, so that $|B| = 2^m$.
- **Hybrid.** Use clusters but choose within-cluster vertices that are geometrically close on the hypercube.

For theory, we can simply assume blocks of size $k = k(N)$ with k polylogarithmic in N. For experiments, we can choose practical sizes (e.g., 8–64 vertices per block).

5.3 Statistical tunneling conjecture

With the planted model from Section 3 and block moves defined, we aim for a result of the following form.

Conjecture 2 (Statistical tunneling for planted hypercube QAP). Under the planted pseudo-Gray model with subgaussian noise and suitable margin conditions, there exist constants $\varepsilon > 0$, $c > 0$, and a block size function $k(N) = O(\log^c N)$ such that, with high probability over the instance:

1. **Near-optimality of block-local minima.** Any assignment π that is a local minimum under block moves of size at most $k(N)$ satisfies
$$f(\pi) - f(\pi^) \leq \varepsilon N. \tag{19}$$
2. **Polynomial-time convergence.** A randomized local-search algorithm that alternates 2-swap hill-climbing and randomized block reoptimizations, starting from a random initialization, reaches an assignment $\hat{\pi}$ with
$$f(\hat{\pi}) - f(\pi^) \leq \varepsilon N \tag{20}$$
in time polynomial in N with probability at least $1 - e^{-\alpha N}$ for some $\alpha > 0$.

Intuitively, this says that deep bad local minima are rare in typical planted instances once we allow block moves of modest size; any persistent local minimum is near-globally optimal.

5.4 Strategy toward a proof

A plausible roadmap is:

1. **Energy gap as a function of Hamming distance in S_N.** Show that if π differs from π on a fraction ρ of vertices, then
$$f(\pi) - f(\pi) \geq \Delta(\rho) N, \tag{21}$$
where $\Delta(\rho)$ is increasing in ρ and positive for $\rho > 0$, with high probability.
2. **Existence of improving blocks for "far" assignments.** Show that if $\rho(\pi, \pi)$ (the fraction of mismatched vertices) is above some threshold, then there exists a block B of size $\leq k(N)$ such that reassigning vertices in B according to π decreases the objective by at least some $\gamma > 0$.
3. **Absence of far block-local minima.** Conclude that any block-local minimum cannot have $\rho > \rho_0$; otherwise, an improving block would contradict local minimality.
4. **Refinement to near-optimality.** Show that assignments with small ρ are themselves ε-optimal in cost, due to Lipschitz-like behavior of the objective with respect to permutations.
5. **Algorithmic convergence.** Analyze the randomized block-selection procedure as a Markov chain whose drift in ρ is negative when ρ is large; argue polynomial hitting time to the region $\rho \leq \rho_0$.

6 Gray-Tunneled Hashing: Algorithmic Perspective

We now translate the theoretical structure into an algorithmic approach for Gray-Tunneled Hashing usable in actual vector databases.

6.1 Objective for real embeddings

For real-world embeddings we typically have $M \ll 2^n$ or $M \gg 2^n$. For simplicity, consider the case where we use a codebook (e.g., cluster centroids or PQ codevectors) of size $N = 2^n$, and actual points are assigned to nearest codebook vectors.

In that setting, our primary assignment is between N codebook vectors and 2^n binary codes. The objective becomes

$$f(\pi) = \sum_{(u,v)\in E(Q_n)} \|c_{\pi(u)} - c_{\pi(v)}\|^2, \tag{22}$$

where c_i are codebook vectors. We want to minimize $f(\pi)$, leveraging 2-swap and block moves, yielding a pseudo-Gray mapping from indices to codes.

6.2 Algorithm sketch

A practical algorithm for Gray-Tunneled Hashing could proceed as:

1. **Initialization.** Obtain codebook vectors $c_1, \ldots, c_N$. Construct an initial assignment π_0, e.g.: sort codebooks by a principal component and assign in Gray-code order; or use any heuristic mapping respecting some notion of locality.

2. **2-swap local optimization.** Run a local-search procedure that repeatedly applies improving 2-swaps on π until no improvement is possible (or until a budget is reached).

3. **Block tunneling.** Identify candidate blocks: groups of codebooks that are close in embedding space or in current code space. For each block, solve the small QAP restricting to that block and apply the best improving reassignment. Iterate between 2-swap local search and block tuning.

4. **Embedding assignment and indexing.** Once a good π is found, assign each codebook index to a binary code according to π. For each embedding x, compute its nearest codebook vector c_i, and thus its binary code via $\pi^{-1}(i)$. Store binary codes in an ANN index (e.g., a Hamming-based index).

5. **ANN queries.** For a query embedding q, compute its code and perform Hamming-distance search. Optionally re-rank top candidates using float embeddings.

6.3 Expected benefits

Compared to naive binary coding, we expect:

- Better locality: Hamming-1 neighbors correspond more consistently to nearest codebooks, reducing semantic distortion for near codes.

- Higher recall at same bit-length: because the code geometry is better aligned with embedding geometry, Hamming balls around a query should contain more true neighbors.

- Or fewer bits for same quality: pseudo-Gray structure may allow using shorter codes (fewer bits) while maintaining a given target recall, reducing memory.

All this is achieved without changing the underlying Hamming-based infrastructure of vector databases.

7 Research and Experimental Program

We now summarize the concrete research program implied by the theory.

7.1 Phase I – Synthetic landscapes

- Implement synthetic planted QAP instances with known π^*.
- Measure: distribution of local minima under 2-swap; effect of block moves on reaching near-π assignments.
- Explore how noise variance and margins affect the depth of bad local minima.

7.2 Phase II – Real embeddings and code assignment

- Use real ANN datasets and precomputed embeddings.
- Learn codebooks (e.g., k-means, PQ) and then optimize code assignment with Gray-Tunneled Hashing.
- Benchmark against: sign hashing; off-the-shelf binary quantization; random code assignments.
- Report recall@k vs bits vs latency.

7.3 Phase III – End-to-end learning

- Incorporate pseudo-Gray regularization and code assignment optimization into autoencoders or VQ-VAE frameworks.
- Evaluate downstream performance in retrieval and robustness.

8 Conclusion

We presented a conceptual and partially formal framework for Gray-Tunneled Hashing, grounded in: a hypercube QAP formulation of code assignment; an elementary landscape result under 2-swap with explicit eigenvalue $\lambda = 4/N$; a planted pseudo-Gray model with subgaussian noise; and a statistical tunneling conjecture via block moves.

The theoretical and algorithmic program aims at turning code assignment from a heuristic detail into a controlled combinatorial optimization problem, with direct implications for vector databases and large-scale retrieval. The next steps are to formalize the planted model and prove a first nontrivial tunneling theorem for block operators, and to implement and benchmark Gray-Tunneled Hashing in realistic vector database setups.

References

[1] D. Whitley. Elementary landscapes. In *Proceedings of GECCO*, 2008.

[2] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.