ORDER MATTERS: A FINITE-CAPACITY MODEL FOR SELECTING INSERTION ORDER IN GRAPH-BASED MAXIMUM INNER PRODUCT SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph-based indexes for maximum inner product search (MIPS) are constructed incrementally, yet the effect of insertion order on the resulting graph is largely overlooked. We show that insertion order can substantially change search performance even when the graph algorithm and all of its parameters remain fixed, and study three schedules: increasing norm, decreasing norm, and random insertion. A stylized finite-capacity model relates their behavior to two geometric properties of the base vectors. The first measures directional redundancy within candidate sets matched to the neighbor-list size, while the second measures how three-vector concentration changes along the norm ranking beyond what pairwise concentration explains. Random projections estimate both quantities before construction and yield a closed-form rule for selecting an order. At matched recall, the selected order achieves about $1.4\times$ query throughput over the better of the other two schedules. Our code is available at `https://anonymous.4open.science/r/artifact-7f3c91/`.

1 INTRODUCTION

Maximum inner product search (MIPS) is a fundamental operation in large-scale retrieval and recommendation. Given a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^d$ and a query $\boldsymbol{q} \in \mathbb{R}^d$, top-$k$ MIPS returns the k vectors with the largest values of $\boldsymbol{q}^\top \boldsymbol{x}_i$. Write each nonzero data vector as $\boldsymbol{x}_i = r_i \boldsymbol{u}_i$, where $r_i = \|\boldsymbol{x}_i\|_2$ is its norm and $\boldsymbol{u}_i \in \mathbb{S}^{d-1}$ is its direction. The score $\boldsymbol{q}^\top \boldsymbol{x}_i = \|\boldsymbol{q}\|_2 r_i \cos \angle(\boldsymbol{q}, \boldsymbol{u}_i)$ depends jointly on vector norm and angular alignment with the query. Exhaustively evaluating this score for every data vector is too expensive at scale, motivating approximate indexes for MIPS.

Currently, approximate MIPS indexes include hashing, tree-based search, quantization, and graph methods (Yan et al., 2018; Koenigstein et al., 2012; Guo et al., 2020; Morozov & Babenko, 2018; Lu et al., 2026). Among them, graph indexes remain one of the most competitive practical choices. A graph index represents each data vector as a vertex with a bounded list of outgoing neighbors. During construction, candidate neighbors are found for each vertex and pruned to fit this list. At query time, search starts from one or more entry vertices and follows edges toward vectors with larger inner products with the query. Search quality therefore depends on which candidates survive pruning under the limited neighbor-list capacity.

Among the many approaches to graph construction, two competitive MIPS indexes, ip-NSW (Morozov & Babenko, 2018) and PAG (Lu et al., 2026), build their graphs incrementally by inserting one data vector at a time. Let $\sigma = (\sigma_1, \ldots, \sigma_N)$ denote a concrete permutation of the vector IDs. When $\boldsymbol{x}_{\sigma_t}$ is inserted, candidate search and pruning can use only the graph formed by the preceding prefix $\{\boldsymbol{x}_{\sigma_j} : j < t\}$. A neighbor that is unavailable or discarded at this step may not be recovered later, so different insertion orders can produce different graphs even when the builder and all of its parameters are fixed, consistent with insertion-order sensitivity previously observed for HNSW (Elliott & Clark, 2024). The angular term in the MIPS score changes with the query, whereas the norm r_i is intrinsic to each vector and strongly shapes MIPS results: high-norm vectors are substantially more likely to appear among the top results (Liu et al., 2020). Norm therefore provides a natural axis for comparing insertion orders. We study three insertion orders: ASC inserts vectors from small to large norm, DESC inserts them from large to small norm, and RAND is a uniformly random permutation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

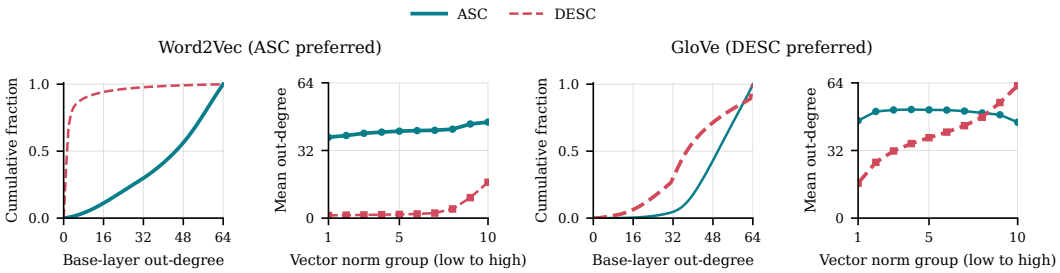


Figure 1: Impact of ASC and DESC. For each dataset, the left panel shows the cumulative distribution: the x axis is the out-degree, and the y axis is the fraction of vectors whose out-degree does not exceed that value. The right panel divides vectors into ten equal-size groups by norm, ordered from the smallest norms (1) to the largest (10), and reports the mean out-degree in each group.

Thus, ASC and DESC are the two extremes of norm ordering, while RAND favors neither extreme. Our goal is to predict efficiently, before graph construction, which of these three schedules will yield the best search performance for a given dataset. Notably, selecting an order by trial requires repeated graph construction and search performance evaluation, which is too costly for practical use (Appendix B.8). We therefore seek a criterion computed directly from the base vectors that captures the geometry of data distribution.

Before proceeding, we first show the effect of insertion order on the constructed graph. Figure 1 reports the final base-layer out-degree of ip-NSW graphs with a base-layer storage limit of 64, built under ASC and DESC on Word2Vec and GloVe-200. On Word2Vec, ASC gives better search performance and produces much fuller neighbor lists: its median out-degree is 45, whereas the median under DESC is 2, and 97.6% of the neighbor lists under DESC use less than half of the available capacity. GloVe-200 exhibits the opposite preference: DESC gives better search performance even though its graph is sparser on average and allocates progressively more edges to vectors with larger norms. These opposite preferences suggest a common mechanism: insertion order changes how pruning distributes the limited positions in each neighbor list across directions. If strong high-norm candidates cover many distinct directions, exposing them early can establish useful routes before the lists fill, favoring DESC. If the same candidates repeatedly represent similar directions, they compete for the same positions, and delaying them can preserve more alternatives, favoring ASC. After the first neighbor is retained, each pruning decision also involves the vector whose list is being built, a new candidate, and an already retained neighbor. Notably, this three-vector interaction possibly makes RAND preferable to both norm-sorted schedules. A model of insertion order must therefore explain both which norm-sorted schedule is favored and when sorting itself is harmful.

In this paper, to tackle the above issues, we design a finite-capacity model of the bounded neighbor lists produced by pruning. The model assigns a common utility to the three schedules: Δ_e distinguishes ASC from DESC, while Δ_s determines whether RAND can outperform the better sorted order. This utility yields a closed-form rule for choosing among the three schedules. We compute both coefficients directly from the base vectors. For Δ_e, uniformly sampled directions identify strong high-inner-product candidates, whose average squared cosine similarity measures directional redundancy at a scale set by the neighbor-list capacity. For Δ_s, nested random hyperplanes divide the unit sphere into progressively finer regions, and we compare how often pairs and triples of directions fall in the same region as the low- and high-norm prefixes grow. The neighbor-list capacity determines both the candidate-pool size used for Δ_e and the hyperplane resolution used for Δ_s. For fixed numbers of sampled directions and hyperplanes, the computation is linear in data size.

Our contributions are threefold. (1) We show that insertion order alone can substantially change both the structure and search performance of an incremental MIPS graph. With the builder and all parameters fixed, different orders redistribute edges across norm levels, and no single order is uniformly best. (2) We propose a stylized finite-capacity model for ASC, DESC, and RAND. The model connects the first and later pruning decisions to two geometric properties of the data and yields a closed-form rule for selecting among the three orders. (3) We turn this rule into an efficient selector using random directions and nested random hyperplanes. In experiments on twelve datasets with ip-NSW and PAG, the selected order matches the best of the three measured schedules in

every condition sharing a recall interval at the default graph degree and achieves about $1.4\times$ query throughput at matched recall over the better of the other two schedules.

2 Related Work

Existing MIPS indexes span hashing, tree-based, quantization, and graph approaches. Hashing methods include ALSH, Simple-LSH, Norm-Ranging LSH, and H2-ALSH (Shrivastava & Li, 2014; Neyshabur & Srebro, 2015; Yan et al., 2018; Huang et al., 2018). Tree-based methods include cone trees, recommendation trees, ProMIPS, and LRUS-CoverTree (Ram & Gray, 2012; Koenigstein et al., 2012; Song et al., 2021; Ma et al., 2024). Quantization methods include QUIP, NEQ, ScaNN, and SOAR (Guo et al., 2016; Dai et al., 2020; Guo et al., 2020; Sun et al., 2023). Our work is most closely related to graph indexes. General-purpose methods such as NSW, HNSW, NSG, and SSG navigate sparse proximity graphs under metric or angular similarity (Malkov et al., 2014; Malkov & Yashunin, 2020; Fu et al., 2019; 2022). For MIPS, Morozov and Babenko introduced ip-NSW, and Liu et al. later showed that large-norm vertices tend to acquire high in-degree and dominate graph traversal, motivating the angular graph used by ip-NSW+ (Morozov & Babenko, 2018; Liu et al., 2020). Other MIPS graph indexes include Möbius-Graph, which transforms MIPS into graph search; NAPG, which adjusts graph proximity using vector norms; PSP and MAG, which exploit connections between inner product and Euclidean geometry; and PAG, which augments graph traversal with projection information (Zhou et al., 2019; Tan et al., 2021; Chen et al., 2025b;a; Lu et al., 2026). Our degree analysis extends this line of work by examining how insertion order and pruning redistribute retained outgoing edges across norm levels.

As for research on the effect of insertion order, Elliott and Clark have shown that the recall of standard HNSW (Malkov & Yashunin, 2020) varies under insertion orders based on pointwise local intrinsic dimensionality (LID) (Elliott & Clark, 2024). We instead study MIPS, where norms interact directly with candidate generation and pruning, and predict among increasing-norm, decreasing-norm, and random insertion before construction. Notably, this intervention is distinct from graph reordering for cache locality (Coleman et al., 2022): such methods change the memory layout of an already constructed graph while preserving its topology, whereas insertion order changes the prefixes available during construction and can therefore change the graph itself.

3 Insertion Order under Finite Capacity

This section develops a finite-capacity model for comparing the three insertion orders under a bounded neighbor-list capacity. We first define a common utility from the pruning rule, then compare Asc with Desc and random insertion with their midpoint.

3.1 Pruning and the finite-capacity utility

Let us first review the standard pruning rule adopted by incremental builders. Consider the neighbor list being constructed for $\boldsymbol{x} = r_x\boldsymbol{u}$, a candidate $\boldsymbol{y} = r_y\boldsymbol{v}$, and a neighbor $\boldsymbol{z} = r_z\boldsymbol{w}$ already retained in the list, where $r_x, r_y, r_z > 0$ and $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{S}^{d-1}$. The inner-product pruning rule discards $\boldsymbol{y}$ when $\boldsymbol{y}^\top\boldsymbol{z} \geq \boldsymbol{y}^\top\boldsymbol{x}$. Substituting the norm-direction decompositions and cancelling r_y gives

$$r_z\boldsymbol{v}^\top\boldsymbol{w} \geq r_x\boldsymbol{v}^\top\boldsymbol{u}. \tag{1}$$

The norm of $\boldsymbol{y}$ affects candidate selection and pruning differently. Before pruning, a larger r_y can increase $\boldsymbol{x}^\top\boldsymbol{y} = r_x r_y\boldsymbol{u}^\top\boldsymbol{v}$, making $\boldsymbol{y}$ more likely to enter the candidate set. In the pruning test above, r_y appears on both sides and therefore cancels. The decision then depends on the three directions $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$, together with the norm ratio r_z/r_x. We call a retained neighbor that causes a candidate to be discarded a blocker. Only previously inserted vectors are available as candidates or blockers. Insertion order therefore changes both which candidates are considered and which candidates survive pruning.

The pruning rule distinguishes the first retained neighbor from those that follow. The first neighbor enters before any blocker has been retained, whereas every later candidate is evaluated in the presence of retained neighbors. The first position and the later positions therefore enter the comparison

differently. We use the first position to compare ASC with DESC, and the later positions to compare RAND with the average of the two sorted orders.

We model one pruning call with a list of $S \geq 2$ positions. Let $R_\pi \in [0,1]$ be the first-position score of order π, and let $L_{\text{ASC}}, L_{\text{DESC}} \in [0,1]$ denote the later-position scores of the two sorted orders. We write $L_{\text{mid}} = (L_{\text{ASC}} + L_{\text{DESC}})/2$ for their midpoint and let $L_{\text{RAND}} \in [0,1]$ be the corresponding score for random insertion. The model retains the part of later-position behavior that is symmetric when ASC and DESC are exchanged, assigning L_{mid} to both sorted endpoints. Its utilities are

$$\begin{aligned} U_{\text{ASC}} &= \frac{R_{\text{ASC}} + (S-1)L_{\text{mid}}}{S}, \qquad U_{\text{DESC}} = \frac{R_{\text{DESC}} + (S-1)L_{\text{mid}}}{S}, \\ U_{\text{RAND}} &= \frac{R_{\text{RAND}} + (S-1)L_{\text{RAND}}}{S}. \end{aligned} \tag{2}$$

We write $\eta = (U_{\text{DESC}} - U_{\text{ASC}})/2$ and $\kappa = U_{\text{RAND}} - (U_{\text{ASC}} + U_{\text{DESC}})/2$. The sign of η selects between ASC and DESC, while RAND has the largest utility exactly when $\kappa > |\eta|$. The difference between the two sorted orders is represented by R_π, while L_{mid} is their average later-position score. Retaining the difference $L_{\text{DESC}} - L_{\text{ASC}}$ would replace η by $\eta_{\text{full}} = [R_{\text{DESC}} - R_{\text{ASC}} + (S-1)(L_{\text{DESC}} - L_{\text{ASC}})]/(2S)$. Appendix B.4 measures this difference and shows that it is substantial. Section 3.2 derives the first-position scores R_π, Section 3.3 derives L_{mid} and L_{RAND}, and Section 3.4 combines the two comparisons.

3.2 THE ENDPOINT COMPARISON

The first-position comparison determines whether large-norm candidates should be exposed early or delayed. Its preferable endpoint depends on how much directional diversity a list of size S can preserve. We define this local competition by comparing two feasible lists of size S: one contains the retained candidates and the other contains alternatives. A one-for-one exchange between the lists is determined by their pooled support, which contains at most $2S$ distinct candidates. For this exchange, we draw a direction $\boldsymbol{\omega}$ uniformly from $\mathbb{S}^{d-1}$. Along this direction, vectors are ranked by their inner products $\boldsymbol{\omega}^\top \boldsymbol{x}$. Let $\mathcal{P}_{\boldsymbol{\omega}}$ contain the $2S$ vectors with the largest such values, representing the candidates most strongly exposed along $\boldsymbol{\omega}$. For any nonzero vector $\boldsymbol{x}$, write $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{x}/\|\boldsymbol{x}\|_2$ for its direction. Then we draw an ordered pair $(\boldsymbol{x}, \boldsymbol{x}')$ uniformly without replacement from $\mathcal{P}_{\boldsymbol{\omega}}$, and define their average squared cosine by

$$c_{\boldsymbol{\omega}} = \mathbb{E}\Big[\big(\boldsymbol{u}(\boldsymbol{x})^\top \boldsymbol{u}(\boldsymbol{x}')\big)^2 \mid \boldsymbol{\omega}\Big] = \frac{1}{2S(2S-1)} \sum_{\substack{\boldsymbol{y},\boldsymbol{y}' \in \mathcal{P}_{\boldsymbol{\omega}} \\ \boldsymbol{y} \neq \boldsymbol{y}'}} \big(\boldsymbol{u}(\boldsymbol{y})^\top \boldsymbol{u}(\boldsymbol{y}')\big)^2. \tag{3}$$

The quantity $c_{\boldsymbol{\omega}}$ measures directional redundancy within the candidate pool. Parallel directions contribute one, orthogonal directions contribute zero, and intermediate angles contribute continuously through their squared cosine. Sampling without replacement keeps the two candidates distinct, matching the pruning comparisons in Section 3.1. Appendix A.1 gives the quadratic characterization of this redundancy measure. In our exchange model, we define the redundancy of the sampled candidates $\boldsymbol{x}$ and $\boldsymbol{x}'$ by $\rho = \big(\boldsymbol{u}(\boldsymbol{x})^\top \boldsymbol{u}(\boldsymbol{x}')\big)^2$. The value ρ is the redundant fraction, while $1-\rho$ is the additional directional coverage contributed by the second candidate. Accordingly, the occupied position loses the share ρ, while the complementary share $1-\rho$ is distributed uniformly among the other $S-1$ positions. The alternative position involved in the exchange therefore receives $(1-\rho)/(S-1)$. Averaging this rule over candidate pairs replaces ρ by $c_{\boldsymbol{\omega}}$, so the net gain from exposing large-norm candidates first is

$$-c_{\boldsymbol{\omega}} + \frac{1 - c_{\boldsymbol{\omega}}}{S-1} = \frac{S}{S-1}\left(\frac{1}{S} - c_{\boldsymbol{\omega}}\right). \tag{4}$$

The balance is positive when the candidates are sufficiently diverse for early exposure to be useful, and negative when their directions are sufficiently redundant for delaying them to preserve more alternatives. Since its sign is determined by $1/S - c_{\boldsymbol{\omega}}$, we define

$$\Delta_e = \mathbb{E}_{\boldsymbol{\omega}}\left[\operatorname{sgn}\left(\frac{1}{S} - c_{\boldsymbol{\omega}}\right)\right]. \tag{5}$$

For each $\boldsymbol{\omega}$, the local vote is $+1$ when the balance favors DESC, -1 when it favors ASC, and zero at a tie. Hence Δ_e is the difference between the fractions of directions favoring the two endpoints, with ties split equally. The corresponding first-position scores are

$$R_{\text{DESC}} = \frac{1+\Delta_e}{2}, \qquad R_{\text{ASC}} = \frac{1-\Delta_e}{2}, \qquad \text{and} \qquad R_{\text{RAND}} = \frac{1}{2}. \tag{6}$$

The sign of Δ_e therefore selects the preferred norm endpoint.

3.3 POSITIONS DECIDED BY PRUNING

We now construct the later-position scores L_{mid} and L_{RAND}, whose difference determines whether random insertion can outperform the two sorted endpoints. Each pruning comparison for such a position involves three vectors: the vector whose neighbor list is being constructed, a candidate, and a retained neighbor. This three-vector dependence is essential because any utility additive over pairwise arrival terms places a random permutation at the midpoint of the two reversed norm orders. Appendix A.2 proves this property.

We measure this three-vector structure using random hyperplane collisions. Let $H_b(\boldsymbol{u}) \in \{-1,+1\}^b$ record the signs of a unit direction $\boldsymbol{u}$ with respect to b independent random hyperplanes. For a fixed realization H, let $C_2(H)$ and $C_3(H)$ be the probabilities that two or three independently sampled directions have the same sign vector, and define $C_{2,b} := \mathbb{E}_H[C_2(H)]$ and $C_{3,b} := \mathbb{E}_H[C_3(H)]$. If p_b is the probability mass of the region containing a direction sampled from the data distribution, then $\mathbb{E}[p_b] = C_{2,b}$ and $\mathbb{E}[p_b^2] = C_{3,b}$. We therefore define

$$T_b := \frac{C_{3,b}}{C_{2,b}^2} = \frac{\mathbb{E}[p_b^2]}{\mathbb{E}[p_b]^2} = 1 + \frac{\mathrm{Var}(p_b)}{\mathbb{E}[p_b]^2}. \tag{7}$$

Thus, T_b measures the variation in directional density across the hyperplane regions. Its Rényi entropy interpretation is given in Appendix A.2.

For a finite set $\mathcal{X}$ of vectors, we define $C_{2,b}(\mathcal{X})$ and $C_{3,b}(\mathcal{X})$ analogously by sampling two or three distinct vectors without replacement and averaging their collision probabilities over the hyperplanes. A collision probability is set to zero when $\mathcal{X}$ contains fewer vectors than required. We then set $T_b(\mathcal{X}) := C_{3,b}(\mathcal{X})/C_{2,b}(\mathcal{X})^2$. Increasing b refines the angular partition and reduces the expected region occupancy. In particular, $(N-1)C_{2,b}(\mathcal{D})$ is the expected number of other vectors sharing the region of a uniformly sampled vector. Since the model in Section 3.2 compares two lists of size S, we choose the largest depth whose expected occupancy remains at least $2S$:

$$b_S = \max\left\{b \in \mathbb{Z}_{\geq 0} : (N-1)C_{2,b}(\mathcal{D}) \geq 2S\right\}. \tag{8}$$

Having fixed b_S, we examine how the directional structure of the inserted vectors changes during construction. For $\tau \in (0,1]$, let $m_\tau = \lceil \tau N \rceil$ be the number of vectors present at construction fraction τ. Under random insertion, these vectors form a uniformly sampled subset of $\mathcal{D}$. Random insertion therefore introduces no systematic dependence between construction progress and norm rank, and we use $T_{b_S}(\mathcal{D})$ as its reference value. For $\pi \in \{\text{ASC}, \text{DESC}\}$, let $\mathcal{D}_\tau^\pi$ denote the first m_τ vectors under order π, corresponding to the smallest-norm fraction τ for ASC and the largest-norm fraction τ for DESC. We refer to $\{\mathcal{D}_\tau^\pi\}_{\tau\in(0,1]}$ as the norm-ranked prefix sequence of π. We compare each prefix with the full dataset using the bounded relative difference below. Appendix A.2 gives the equivalent cross-multiplied form when a collision probability is zero.

$$g_\pi(\tau) = \frac{T_{b_S}(\mathcal{D}_\tau^\pi) - T_{b_S}(\mathcal{D})}{T_{b_S}(\mathcal{D}_\tau^\pi) + T_{b_S}(\mathcal{D})}, \qquad \Delta_s = \frac{1}{2}\int_0^1 \left(g_{\text{ASC}}(\tau) + g_{\text{DESC}}(\tau)\right) d\tau. \tag{9}$$

A positive $g_\pi(\tau)$ means that the prefix has greater variation in directional density than the full dataset, while a negative value means that it has less. Integrating over τ covers the complete prefix sequence. Because random insertion is compared with the average of the two norm-sorted orders, Δ_s averages their prefix sequences and is unchanged when ASC and DESC are exchanged. If norm and direction are independent, selecting either the largest- or smallest-norm fraction leaves the directional distribution unchanged. Consequently, $\mathcal{D}_\tau^\pi$ has the same directional distribution as $\mathcal{D}$ for both orders, giving $g_\pi(\tau) = 0$ and $\Delta_s = 0$.

We now use Δ_s to specify the later-position midpoint. Greater directional concentration along the two norm-ranked prefix sequences reduces the chance that pruning preserves a new direction. The signed score $-\Delta_s$ is therefore mapped to $L_{\mathrm{mid}} = (1 - \Delta_s)/2$, with zero mapped to the neutral value $1/2$. For random insertion, we average over the six relative arrival orders of an unordered triple. We call a comparison complete when the smallest-norm vector arrives before or after both remaining vectors, and incomplete when it arrives between them. A complete comparison receives the sorted midpoint score L_{mid}, whereas an incomplete comparison receives the neutral score $1/2$. The six-order calculation uses the relative arrival order of the three vectors. Four of the six equally likely orders are complete and two are incomplete, giving

$$L_{\mathrm{mid}} = \frac{1 - \Delta_s}{2}, \qquad L_{\mathrm{RAND}} = \frac{1}{2} - \frac{\Delta_s}{3}. \tag{10}$$

Thus, $L_{\mathrm{RAND}} - L_{\mathrm{mid}} = \Delta_s/6$. Section 3.4 incorporates this difference into the complete utility, and Appendix A.2 gives the collision identities and the six-order calculation.

3.4 The selection rule

The preceding results determine the two utility comparisons in equation 2. Equation 6 gives $R_{\mathrm{DESC}} - R_{\mathrm{ASC}} = \Delta_e$ and $R_{\mathrm{RAND}} = (R_{\mathrm{ASC}} + R_{\mathrm{DESC}})/2$, while equation 10 gives $L_{\mathrm{RAND}} - L_{\mathrm{mid}} = \Delta_s/6$. Substituting these first-position and later-position differences into equation 2 yields the selection rule. Specifically, we have

$$\eta = \frac{\Delta_e}{2S}, \qquad \kappa = \frac{S-1}{6S}\Delta_s \qquad \text{and} \qquad U_{\mathrm{RAND}} - \max\{U_{\mathrm{ASC}}, U_{\mathrm{DESC}}\} = \frac{(S-1)\Delta_s - 3|\Delta_e|}{6S}. \tag{11}$$

Consequently, random insertion uniquely maximizes the model utility exactly when

$$(S-1)\Delta_s > 3|\Delta_e|. \tag{12}$$

Equality in equation 12 gives a tie between random insertion and the better sorted order. Below the boundary, DESC maximizes the utility when $\Delta_e > 0$, whereas ASC maximizes it when $\Delta_e < 0$. When $\Delta_e = 0$, the two sorted orders tie, and we return ASC.

The factor 3 combines the first-position weight $1/S$ with the later-position gap $\Delta_s/6$ in equation 10.

By the above discussion, we summarize the comparison by the selection margin:

$$\Phi_S = (S-1)\Delta_s - 3|\Delta_e| = 6S\left(U_{\mathrm{RAND}} - \max\{U_{\mathrm{ASC}}, U_{\mathrm{DESC}}\}\right).$$

4 Efficient Order Selection

The coefficients in Section 3 are expectations over random directions and hyperplanes. We estimate them from the base vectors and use the resulting order to construct the graph.

4.1 Estimating the coefficients

For Δ_e, we sample L_e independent normalized Gaussian directions together with their negatives. A projection scan finds the $2S$ vectors in $\mathcal{P}_{\boldsymbol{\omega}}$ for each direction. If $G_{\boldsymbol{\omega}}$ is the Gram matrix of their normalized directions, the distinct-vector redundancy is

$$\widehat{c}_{\boldsymbol{\omega}} = \frac{\|G_{\boldsymbol{\omega}}\|_F^2 - 2S}{2S(2S-1)}. \tag{13}$$

For each sampled direction, we average $\mathrm{sgn}(1/S - \widehat{c}_{\boldsymbol{\omega}})$ with the corresponding value for its negative. Averaging these values over the L_e sampled directions gives $\widehat{\Delta}_e$. For Δ_s, we draw B_H independent sequences of Gaussian hyperplanes. The first b signs in each sequence identify the region at depth b, giving all depths in one nested construction. For an integer z and $\nu \in \{2, 3\}$, write $(z)_\nu = z(z-1)\cdots(z-\nu+1)$. If a set $\mathcal{X}$ contains m vectors and its region occupancies are $(n_j)_j$, we set $\widehat{C}_{\nu,b}(\mathcal{X}) = 0$ when $m < \nu$. Whenever the denominator is positive, the corresponding distinct-vector collision estimate is

$$\widehat{C}_{2,b}(\mathcal{X}) = \frac{\sum_j (n_j)_2}{(m)_2}, \qquad \widehat{C}_{3,b}(\mathcal{X}) = \frac{\sum_j (n_j)_3}{(m)_3}. \tag{14}$$

Algorithm 1 Insertion-order selection

Require: Base vectors $\mathcal{D}$, list size S
Ensure: Insertion order $\pi \in \{\text{ASC}, \text{DESC}, \text{RAND}\}$

1: Form the increasing- and decreasing-norm rankings ▷ *Section 3*
2: Estimate $\widehat{\Delta}_e$ from the top-$2S$ projected sets over L_e Gaussian directions and their negatives ▷ *equations 13 and 5*
3: Estimate the full-data pair collisions using B_H nested hyperplane sequences ▷ *equation 14*
4: Determine the hyperplane depth $\widehat{b}_S$ ▷ *equation 15*
5: Scan both norm rankings at depth $\widehat{b}_S$ to estimate $\widehat{\Delta}_s$ ▷ *equations 7 and 9*
6: $\widehat{\Phi}_S \leftarrow (S-1)\widehat{\Delta}_s - 3|\widehat{\Delta}_e|$ ▷ *Section 3.4*
7: $\pi \leftarrow \begin{cases} \text{RAND}, & \widehat{\Phi}_S > 0, \\ \text{DESC}, & \widehat{\Phi}_S \le 0 \text{ and } \widehat{\Delta}_e > 0, \\ \text{ASC}, & \text{otherwise.} \end{cases}$
8: **return** π

Averaging over the B_H sequences gives $\overline{C}_{2,b}$ and $\overline{C}_{3,b}$. Replacing the exact collision probabilities in equation 8 by these averages gives

$$\widehat{b}_S = \max\left\{b \in \mathbb{Z}_{\ge 0} : (N-1)\overline{C}_{2,b}(\mathcal{D}) \ge 2S\right\}. \tag{15}$$

At this depth, we set $\widehat{T}_{\widehat{b}_S}(\mathcal{X}) = \overline{C}_{3,\widehat{b}_S}(\mathcal{X})/\overline{C}_{2,\widehat{b}_S}(\mathcal{X})^2$ whenever the denominator is positive. When a collision count is zero, we evaluate the relative difference in equation 9 using the cross-multiplied form in Appendix A.2. A single scan of the ASC and DESC rankings updates these counts for every prefix and yields $\widehat{\Delta}_s$. Together, $\widehat{\Delta}_e$ and $\widehat{\Delta}_s$ determine the insertion order through the rule in Section 3.4 and Algorithm 1 summarizes the complete procedure.

4.2 Estimator consistency and complexity

For each sampled Gaussian direction, the direction and its negative produce one signed value in $[-1, 1]$, and averaging these values gives $\widehat{\Delta}_e$. For $\widehat{\Delta}_s$, each independent hyperplane sequence produces pair and triple collision probabilities in $[0, 1]$. Averaging these probabilities and applying equation 9 gives $\widehat{\Delta}_s$. As L_e and B_H increase, the two estimates converge almost surely to Δ_e and Δ_s. If the threshold in equation 8 is crossed strictly and the largest model utility is unique, substituting the estimates into the selection rule eventually gives the same order as using Δ_e and Δ_s. Appendix A.3 proves the result, and Appendix B.2 evaluates stability across values of L_e, B_H, and the random seeds.

As for the complexity, computing the two norm rankings costs $O(N \log N)$. Estimating Δ_e requires one projection scan for each of the L_e sampled directions, for a total cost of $O(NdL_e)$. Notably, the scans are independent and can be performed in parallel. Estimating Δ_s costs $O(NdB_H b_S)$, and the B_H hyperplane sequences can likewise be processed in parallel. Once the hyperplane signs have been computed, adding the next vector along either norm ranking changes only one region count, so the pair and triple collision totals require constant work per vector and hyperplane sequence. The total cost of selecting an order is therefore $O(N \log N + Nd(L_e + B_H b_S))$.

5 Experiments

All experimental results were obtained on the same Linux server equipped with a 2.00 GHz Intel Xeon Gold 6330 CPU, 28 physical cores, 56 hardware threads, and 256 GB of memory. Graph construction used all 56 hardware threads, while query processing was single-threaded.

We evaluate twelve MIPS datasets: Cohere, DINOv2, FB-SSNPP, Gaussian, GloVe-200, GloVe-300, ImageNet, SproosV3, Tiny5M, WikiGiga, Word2Vec, and ZZSN. The main paper reports ip-NSW and PAG under ASC, DESC, and RAND. Appendix B.7 additionally compares ip-NSW and PAG with Möbius-Graph, ScaNN, SOAR, MAG, and PSP. Dataset details, complete baseline configurations, and additional results are given in Appendix B.

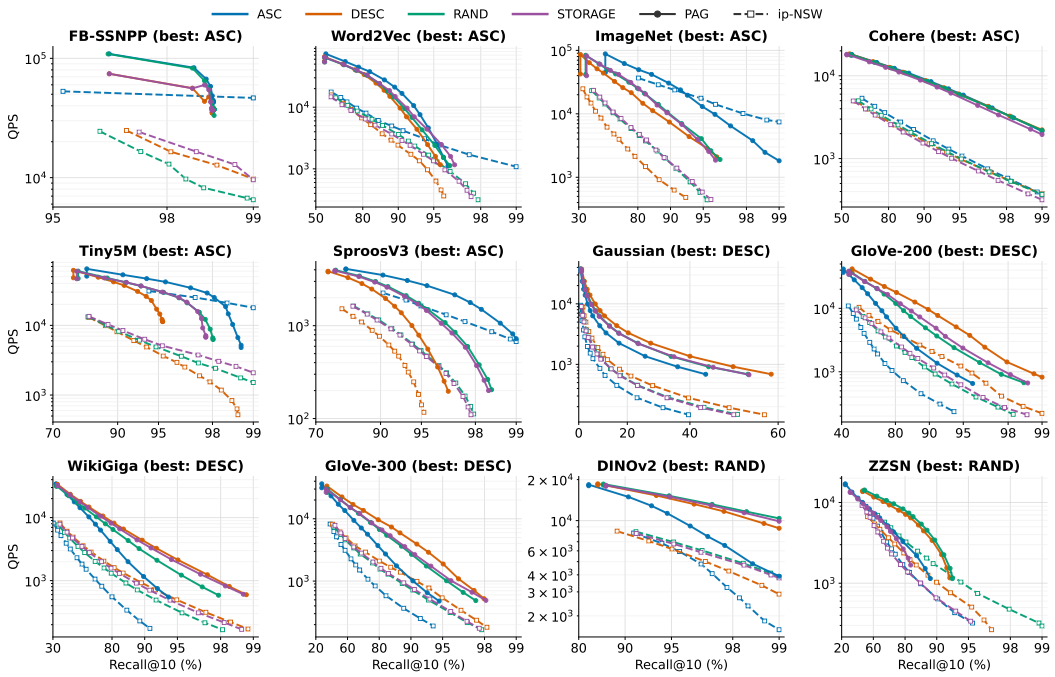


Figure 2: Queries per second (QPS) versus recall at $k = 10$ for ip-NSW and PAG under ASC, DESC, RAND, and the original storage order. Construction and search parameters are fixed within each method. The RAND curve averages five independently sampled insertion orders, while the storage-order curve uses the vector order in the released base file.

For the selector, we set $S = M = 32$, where M denotes the graph degree used by both ip-NSW and PAG. We estimate the coefficients using $L_e = 64$ sampled Gaussian directions together with their negatives and $B_H = 8$ independent hyperplane sequences. The selector also uses 56 threads.

5.1 Main Results on Search Performance

Figures 2 and 3 give four main observations. (1) Insertion order visibly changes the QPS–recall curves: six datasets favor ASC, four favor DESC, and DINOv2 and ZZSN favor RAND. Of the 48 dataset–builder–k conditions, 42 have a recall interval shared by all three orders; Appendix B.6 reports the complete $k = 100$ curves. Within these conditions, always using ASC, DESC, or RAND attains the best measured throughput in only 18, 16, and 8 cases, respectively. (2) The original storage order usually follows RAND and therefore cannot substitute for choosing a sorted schedule. (3) The preferred order is consistent across both builders and retrieval depths at $S = M = 32$. Despite their different graph structures, ip-NSW and PAG exhibit the same six/four/two division among ASC, DESC, and RAND. (4) The selector places all twelve datasets in regions consistent with their measured preferences. In these 42 conditions, it matches the best measured schedule and achieves about $1.4\times$ geometric-mean query throughput at matched recall over the better of the other two schedules. Appendices B.2–B.5 report analyses of estimator stability, the decision constant, graph degree, angular scale, pruning, and scalar baselines.

Figure 3 also shows that the selector stays at ratio one in these 42 conditions, whereas every fixed order develops a long loss tail. No single fixed order therefore avoids substantial throughput loss across the twelve datasets. The two datasets that favor RAND also show why the third branch is necessary: a rule restricted to the two sorted endpoints would miss both, whereas using RAND universally performs poorly on many of the remaining datasets. The selector therefore decides both whether norm sorting is appropriate and, when it is, which norm endpoint should be exposed earlier. The left panel also shows how decisively each dataset lies within its selected region through its distance from the two decision boundaries.

Table 1: Construction and prediction time in seconds. Bold marks the fastest insertion order for each builder and dataset.

Method	Order	Cohere	DINOv2	FB-SSNPP	Gaussian	GloVe-200	GloVe-300	ImageNet	SproosV3	Tiny5M	WikiGiga	Word2Vec	ZZSN
ip-NSW	ASC	**319.0**	162.1	**103.6**	**275.6**	**156.9**	**387.3**	161.4	553.5	**694.0**	**183.2**	91.1	163.7
	DESC	362.6	**137.0**	309.7	283.4	191.4	427.9	**130.6**	**443.5**	793.5	225.1	**75.3**	**152.1**
	RAND	365.5	193.0	218.4	280.0	194.0	461.7	173.2	540.3	1020.2	229.4	111.7	232.7
PAG	ASC	93.7	84.2	**239.0**	**71.0**	**60.1**	**131.3**	73.2	297.9	**609.5**	**67.4**	**91.1**	86.1
	DESC	**85.6**	**80.1**	259.1	85.6	82.2	162.5	**46.3**	**211.8**	655.2	76.4	92.9	**78.5**
	RAND	90.8	84.1	256.8	82.9	80.3	164.3	52.1	213.6	668.2	75.8	97.0	80.8
Selector	–	3.4	4.1	4.7	1.8	2.0	6.1	1.9	12.0	18.4	1.9	1.9	4.4

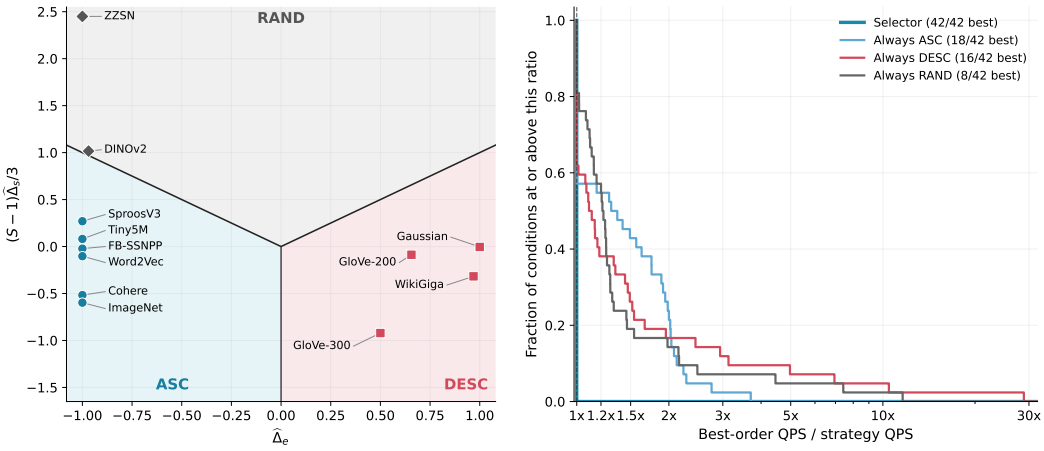


Figure 3: Prediction regions and the cost of using an insertion-order strategy. Left: $\widehat{\Delta}_e$ is plotted horizontally and $(S-1)\widehat{\Delta}_s/3$ vertically; the boundaries select ASC, DESC, or RAND. Right: for the selector and each fixed order, the curve gives the fraction of the 42 conditions defined above in which the best order is at least the indicated factor faster.

5.2 Construction and Prediction Cost

Table 1 compares the selector with complete graph construction. Insertion order also changes construction time. Random insertion is often slower for ip-NSW, while the pattern is less uniform for PAG. The selector takes 1.8–18.4 seconds across the twelve datasets. Relative to construction under the selected order, its median overhead is 1.7% for ip-NSW and 2.8% for PAG, and its maximum overhead remains below 6% for both builders. The computations for $\widehat{\Delta}_e$ and $\widehat{\Delta}_s$ process their sampled directions independently and use the same 56-thread budget as graph construction. The selector runs once before construction, after which the chosen schedule is passed to the graph builder. Its projection scans add a small one-time cost to a single construction pass. Because the coefficients depend only on the base vectors and S, the same prediction and the selector time reported in Table 1 are shared by ip-NSW and PAG.

6 Conclusion

Insertion order can substantially change graph structure and search performance under a fixed constructor. Our finite-capacity model relates this effect to directional redundancy among strong inner-product candidates and to the shift from pairwise to three-vector concentration along the norm ranking. These quantities yield a closed-form rule for selecting increasing-norm, decreasing-norm, or random insertion before construction. Across twelve datasets and two graph builders, the rule matches the measured preference at $S = M = 32$ and delivers about $1.4\times$ geometric-mean query throughput at matched recall over the better of the other two orders. Together, Δ_e and Δ_s turn insertion order into a measurable design choice for incremental MIPS graphs.

AI USE STATEMENT

AI tools were used to assist with language editing, code organization, and figure preparation. All mathematical claims, experimental methodology, figures, and reported results were verified by the authors.

REFERENCES

Tingyang Chen, Cong Fu, Xiangyu Ke, Yunjun Gao, Yabo Ni, and Anxiang Zeng. Stitching inner product and euclidean metrics for topology-aware maximum inner product search. In *SIGIR*, pp. 2341–2350, 2025a.

Tingyang Chen, Cong Fu, Kun Wang, Xiangyu Ke, Yunjun Gao, Wenchao Zhou, Yabo Ni, and Anxiang Zeng. Maximum inner product is query-scaled nearest neighbor. *PVLDB*, 18(6):1770–1783, 2025b.

Benjamin Coleman, Santiago Segarra, Alexander J. Smola, and Anshumali Shrivastava. Graph reordering for cache-efficient near neighbor search. In *NeurIPS*, volume 35, 2022.

Xinyan Dai, Xiao Yan, Kelvin Kai Wing Ng, Jie Liu, and James Cheng. Norm-explicit quantization: Improving vector quantization for maximum inner product search. In *AAAI*, pp. 51–58, 2020.

Wei Dong, Moses Charikar, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *WWW*, pp. 577–586, 2011.

Owen Pendrigh Elliott and Jesse Clark. The impacts of data, ordering, and intrinsic dimensionality on recall in hierarchical navigable small worlds. In *ICTIR*, pp. 25–33, 2024. doi: 10.1145/3664190.3672512.

Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. Fast approximate nearest neighbor search with the navigating spreading-out graph. *PVLDB*, 12(5):461–474, 2019.

Cong Fu, Changxu Wang, and Deng Cai. High dimensional similarity search with satellite system graph: Efficiency, scalability, and unindexed query compatibility. *TPAMI*, 44(8):4139–4150, 2022.

Ruiqi Guo, Sanjiv Kumar, Krzysztof Choromanski, and David Simcha. Quantization based fast inner product search. In *AISTATS*, pp. 482–490, 2016.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *ICML*, pp. 3887–3896, 2020.

Qiang Huang, Guihong Ma, Jianlin Feng, Qiong Fang, and Anthony K. H. Tung. Accurate and fast asymmetric locality-sensitive hashing scheme for maximum inner product search. In *KDD*, pp. 1561–1570, 2018.

Noam Koenigstein, Parikshit Ram, and Yuval Shavitt. Efficient retrieval of recommendations in a matrix factorization framework. In *CIKM*, pp. 535–544, 2012.

Jie Liu, Xiao Yan, Xinyan Dai, Zhirong Li, James Cheng, and Ming-Chang Yang. Understanding and improving proximity graph based maximum inner product search. In *AAAI*, pp. 139–146, 2020.

Kejing Lu, Zhenpeng Pan, Yoshiharu Ishikawa, Chuan Xiao, and Jianbin Qin. Approximate nearest neighbor search for modern AI: A projection-augmented graph approach. In *ICML*, 2026.

Hengzhao Ma, Jianzhong Li, and Yong Zhang. Reconsidering tree based methods for k-maximum inner-product search: The LRUS-CoverTree. In *ICDE*, pp. 4671–4684, 2024.

Yury Malkov, Alexander Ponomarenko, Andrey Logvinov, and Vladimir Krylov. Approximate nearest neighbor algorithm based on navigable small world graphs. *Information Systems*, 45:61–68, 2014.

Yury A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *TPAMI*, 42(4):824–836, 2020.

Stanislav Morozov and Artem Babenko. Non-metric similarity graphs for maximum inner product search. In *NeurIPS*, pp. 4726–4735, 2018.

Behnam Neyshabur and Nathan Srebro. On symmetric and asymmetric LSHs for inner product search. In *ICML*, pp. 1926–1934, 2015.

Parikshit Ram and Alexander G. Gray. Maximum inner-product search using cone trees. In *KDD*, pp. 931–939, 2012.

Anshumali Shrivastava and Ping Li. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *NeurIPS*, pp. 2321–2329, 2014.

Yang Song, Yu Gu, Rui Zhang, and Ge Yu. ProMIPS: Efficient high-dimensional c-approximate maximum inner product search with a lightweight index. In *ICDE*, pp. 1619–1630, 2021.

Philip Sun, David Simcha, Dave Dopson, Ruiqi Guo, and Sanjiv Kumar. SOAR: Improved indexing for approximate nearest neighbor search. In *NeurIPS*, pp. 3189–3204, 2023.

Shulong Tan, Zhaozhuo Xu, Weijie Zhao, Hongliang Fei, Zhixin Zhou, and Ping Li. Norm adjusted proximity graph for fast inner product retrieval. In *KDD*, pp. 1552–1560, 2021.

Xiao Yan, Jinfeng Li, Xinyan Dai, Hongzhi Chen, and James Cheng. Norm-ranging LSH for maximum inner product search. In *NeurIPS*, pp. 2956–2965, 2018.

Zhixin Zhou, Shulong Tan, Zhaozhuo Xu, and Ping Li. Möbius transformation for fast inner product search on graph. In *NeurIPS*, pp. 8216–8227, 2019.

A Model Derivations and Estimator Consistency

This appendix supplies the derivations and estimator-consistency result used in Sections 3 and 4. We follow the order of the main text and refer directly to its definitions.

A.1 The endpoint calculation

Fix a direction $\boldsymbol{\omega}$ and the candidate set $\mathcal{P}_{\boldsymbol{\omega}}$ from Section 3.2. Let $G_{\boldsymbol{\omega}}$ be the Gram matrix of the $2S$ unit directions in this set. Its squared Frobenius norm is the sum of all squared pairwise cosines. The $2S$ diagonal terms are one, while the remaining $2S(2S-1)$ terms correspond to ordered pairs of distinct candidates. Removing the diagonal and averaging the remaining terms gives both the collision identity in equation 3 and the Gram-matrix computation in equation 13.

We next justify the use of squared cosine as the redundancy assigned to a pair of directions. For a fixed unit vector $\boldsymbol{u}$, suppose $\rho(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{v}^\top A_{\boldsymbol{u}} \boldsymbol{v}$ is homogeneous and quadratic in $\boldsymbol{v}$, invariant under a common orthogonal transformation of $\boldsymbol{u}$ and $\boldsymbol{v}$, equal to one when $\boldsymbol{v} = \boldsymbol{u}$, and equal to zero when $\boldsymbol{v} \perp \boldsymbol{u}$. Orthogonal transformations that fix $\boldsymbol{u}$ force $A_{\boldsymbol{u}}$ to have one eigenvalue on $\mathrm{span}(\boldsymbol{u})$ and one on its orthogonal complement. The two boundary values set these eigenvalues to one and zero, respectively. Hence we have

$$\rho(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}^\top \boldsymbol{v})^2.$$

For a candidate pair with redundancy ρ, the candidate-pair balance in Section 3.2 is $-\rho + (1-\rho)/(S-1)$. Averaging over the distinct pairs in $\mathcal{P}_{\boldsymbol{\omega}}$ replaces ρ by $c_{\boldsymbol{\omega}}$ and gives equation 4. Its sign changes at $c_{\boldsymbol{\omega}} = 1/S$, which yields the signed average Δ_e in equation 5. Since each direction contributes a vote in $\{-1, 0, 1\}$, splitting ties equally gives the three first-position scores in equation 6.

A.2 The later-position calculation

The later positions require a three-vector statistic for a structural reason. Let V_σ denote any utility assigned to a concrete permutation σ that is additive over unordered pairs, with each term depending only on which member of the pair arrives first. We write $i \prec_\sigma j$ when record i appears before record j in σ:

$$V_\sigma = a_0 + \sum_{\{i,j\}} \varphi_{ij}(\mathbf{1}\{i \prec_\sigma j\}). \tag{16}$$

The orders Asc and Desc reverse every pair, whereas a uniform random permutation places either member first with probability $1/2$. Taking the expectation of equation 16 term by term therefore gives

$$\mathbb{E}_\sigma V_\sigma = \frac{1}{2}\left(V_{\text{Asc}} + V_{\text{Desc}}\right). \tag{17}$$

Consequently, pairwise arrival effects choose between the two sorted orders, while a strict preference for random insertion requires an interaction among at least three vectors. This is the order of interaction in the pruning comparison among the vector whose list is being built, a candidate, and a retained neighbor.

This observation motivates the pair and triple collision probabilities in Section 3.3. A random-hyperplane partition is rotation invariant and measures pair and triple collisions on the same angular scale. For one hyperplane with normal $\boldsymbol{a}$, write $h_{\boldsymbol{a}}(\boldsymbol{u}) = \mathrm{sgn}(\boldsymbol{a}^\top \boldsymbol{u})$. If θ_{uv} is the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$, rotational invariance gives

$$\Pr\{h_{\boldsymbol{a}}(\boldsymbol{u}) = h_{\boldsymbol{a}}(\boldsymbol{v})\} = 1 - \frac{\theta_{uv}}{\pi}. \tag{18}$$

For three directions, the Gaussian orthant formula gives

$$\Pr\{h_{\boldsymbol{a}}(\boldsymbol{u}) = h_{\boldsymbol{a}}(\boldsymbol{v}) = h_{\boldsymbol{a}}(\boldsymbol{w})\} = 1 - \frac{\theta_{uv} + \theta_{uw} + \theta_{vw}}{2\pi}. \tag{19}$$

Here, π in the denominators denotes the circle constant, distinct from the insertion order π. With b independent hyperplanes, the corresponding probabilities are raised to the b-th power. Thus the nested partitions compare pair and triple concentration in a rotation-invariant form.

The preceding construction measures pair and triple collisions under the same angular partition. However, the raw triple-collision probability also increases with ordinary pairwise concentration. The ratio $T_b = C_{3,b}/C_{2,b}^2$ therefore normalizes the triple collision by the pairwise collision level. To see what remains after this normalization, fix a partition H with region masses $(p_j)_j$. Then $C_2(H) = \sum_j p_j^2$ and $C_3(H) = \sum_j p_j^3$, while Cauchy–Schwarz gives $C_3(H) \geq C_2(H)^2$. With Rényi entropy $\mathrm{H}_\alpha(H) = (1-\alpha)^{-1} \log \sum_j p_j^\alpha$, the same ratio satisfies

$$\log \frac{C_3(H)}{C_2(H)^2} = 2\big(\mathrm{H}_2(H) - \mathrm{H}_3(H)\big). \tag{20}$$

After averaging over the random partitions, Jensen's inequality yields $C_{3,b} \geq C_{2,b}^2$, and the moment calculation in equation 7 shows that $T_b - 1$ is the relative variance of region mass. Hence T_b increases only when the directional density becomes more uneven than its pairwise collision rate alone would indicate.

We next connect this expected collision ratio to the prefixes encountered during construction. For a fixed depth-b partition, each estimator in equation 14 is the fraction of ordered distinct pairs or triples that fall in one region. For $m \geq 3$, let $\mathcal{X}_m$ be a uniformly sampled subset of m vectors from $\mathcal{D}$. Any ordered distinct ν-tuple from $\mathcal{D}$, where $\nu \in \{2,3\}$, belongs to $\mathcal{X}_m$ with probability $(m)_\nu/(N)_\nu$. This factor cancels the denominator in equation 14, and we have

$$\mathbb{E}_{\mathcal{X}_m}\Big[\widehat{C}_{\nu,b}(\mathcal{X}_m) \mid H_b\Big] = \widehat{C}_{\nu,b}(\mathcal{D}), \qquad \nu \in \{2,3\}. \tag{21}$$

Averaging over the hyperplanes gives the same identity for $C_{\nu,b}$. A random prefix therefore has the full-data pair and triple collision probabilities in expectation, which justifies using $T_{b_S}(\mathcal{D})$ as the reference in equation 9. In contrast, the prefixes of ASC and DESC condition on norm rank, and thus their deviation from this reference measures the dependence between norm and directional structure along construction.

Having established the full-data reference, we next fix the angular resolution at which it is compared with the sorted prefixes. The partition at depth $b+1$ refines that at depth b, so a pair that collides at depth $b+1$ must also collide at depth b. Therefore $C_{2,b}(\mathcal{D})$ is nonincreasing in b, and the capacity crossing in equation 8 selects a well-defined depth b_S. At this depth, integrating the two bounded prefix contrasts in equation 9 produces Δ_s. Its sign records whether norm sorting systematically increases or decreases three-way directional concentration relative to the full-data reference.

This completes the definition of Δ_s in terms of expected collision probabilities. Its empirical evaluation requires one additional convention because a prefix may have a zero pair-collision count. We use the cross-multiplied terms

$$J_\pi(\tau) = C_{3,b_S}(\mathcal{D}_\tau^\pi) C_{2,b_S}(\mathcal{D})^2, \qquad K_\pi(\tau) = C_{3,b_S}(\mathcal{D}) C_{2,b_S}(\mathcal{D}_\tau^\pi)^2$$

and define

$$g_\pi(\tau) = \frac{J_\pi(\tau) - K_\pi(\tau)}{J_\pi(\tau) + K_\pi(\tau)}.$$

Whenever both ratios are defined, this is algebraically identical to equation 9 and equals

$$\tanh\left(\frac{1}{2}\log\frac{T_{b_S}(\mathcal{D}_\tau^\pi)}{T_{b_S}(\mathcal{D})}\right).$$

A single zero term is handled by the cross-multiplied expression. When both terms vanish, we assign the neutral value zero.

Finally, Δ_s is converted to the two later-position scores in equation 10. The treatment of complete and incomplete comparisons depends only on the relative arrival order of an unordered triple. It assigns $L_{\mathrm{mid}} = (1-\Delta_s)/2$ when the smallest-norm vector arrives first or last and assigns the neutral score $1/2$ when that vector arrives between the other two. Four of the six relative orders are complete and two are incomplete. Their average is $L_{\mathrm{RAND}} = 1/2 - \Delta_s/3$, so $L_{\mathrm{RAND}} - L_{\mathrm{mid}} = \Delta_s/6$.

A.3 Estimator consistency and incremental updates

For each of the L_e independently sampled normalized Gaussian directions $\boldsymbol{\omega}_\ell$ and its negative, define

$$Z_\ell = \frac{1}{2}\left[\mathrm{sgn}\left(\frac{1}{S} - c_{\boldsymbol{\omega}_\ell}\right) + \mathrm{sgn}\left(\frac{1}{S} - c_{-\boldsymbol{\omega}_\ell}\right)\right].$$

Then $Z_\ell \in [-1,1]$, $\mathbb{E}Z_\ell = \Delta_e$, and the redundancies are evaluated by equation 13. Hoeffding's inequality gives

$$\Pr\{|\widehat{\Delta}_e - \Delta_e| > \epsilon\} \le 2\exp(-L_e\epsilon^2/2).$$

For $\widehat{\Delta}_s$, let H_b^r be the depth-b partition in hyperplane sequence r. For each $\pi \in \{\text{ASC}, \text{DESC}\}$, prefix size m, and $\nu \in \{2,3\}$, the quantity

$$\widehat{C}_{\nu,b}^{\pi,r}(m) = \widehat{C}_{\nu,b}(\mathcal{D}_{m/N}^{\pi}; H_b^r)$$

uses the zero convention from Section 4 when $m < \nu$, lies in $[0,1]$, and has expectation $C_{\nu,b}(\mathcal{D}_{m/N}^{\pi})$. The strong law therefore gives simultaneous convergence of the averages over the finite collection of prefixes and depths used by the selector. Pair and triple probabilities are averaged before their ratio is formed, as required by equation 7.

Proposition 1 (Consistency of the estimated decision)**.** *Suppose the capacity crossing is strict:*

$$(N-1)C_{2,b_S}(\mathcal{D}) > 2S > (N-1)C_{2,b_S+1}(\mathcal{D}). \tag{22}$$

As $L_e, B_H \to \infty$, $\widehat{\Delta}_e \to \Delta_e$, $\widehat{b}_S = b_S$ eventually, and $\widehat{\Delta}_s \to \Delta_s$ almost surely. If the largest model utility is unique, the rule evaluated with the estimates eventually returns the same order as the rule evaluated with Δ_e and Δ_s almost surely.

Proof. The strong law gives $\widehat{\Delta}_e \to \Delta_e$ and joint convergence of every averaged pair and triple collision probability used for the full data and the two norm-ranked prefix sequences. Under equation 22, the estimates eventually preserve both strict inequalities, so $\widehat{b}_S = b_S$.

At the fixed depth b_S, $\widehat{T}_{b_S} \to T_{b_S}$ whenever the pair probability is positive. The cross-multiplied comparison above is continuous in its two nonnegative terms whenever at least one term is positive. If both terms vanish, the comparison formed from the collision probabilities and the corresponding estimated comparison are both assigned the neutral value zero. Averaging over the finite set of prefixes therefore gives $\widehat{\Delta}_s \to \Delta_s$. Consequently,

$$\widehat{\Phi}_S = (S-1)\widehat{\Delta}_s - 3|\widehat{\Delta}_e| \longrightarrow (S-1)\Delta_s - 3|\Delta_e| = \Phi_S.$$

If the largest model utility is unique, its gap from the other two utilities is positive. The convergence of $\widehat{\Phi}_S$ and $\widehat{\Delta}_e$ then implies that the estimated rule eventually returns the same insertion order as the rule evaluated with Δ_e and Δ_s. □

The prefix collision totals admit constant-time updates. If the next vector enters a region containing z vectors, then $(z+1)_\nu - (z)_\nu = \nu(z)_{\nu-1}$. Thus $\sum_j (n_j)_2$ and $\sum_j (n_j)_3$ require constant work per vector and hyperplane sequence along each norm ranking.

B Additional Experiments

B.1 Datasets and Baselines

Datasets. We evaluate twelve MIPS datasets covering language, vision, recommendation, neural activations, and synthetic data. Each dataset contains between one and five million base vectors, with dimensionality ranging from 150 to 4,096. The base and query sets are disjoint, and the ground truth is obtained by exact float32 inner-product search. Table 2 summarizes their statistics.

Baselines. We evaluate seven MIPS methods in total: ip-NSW and PAG with the selected insertion order, together with five graph-based, transformation-based, and quantization-based baselines. We mainly use the authors' implementations and follow their recommended construction settings.

- **ip-NSW** (Morozov & Babenko, 2018) directly builds a navigable small-world graph under inner product. For its parameters, we use $M = 32$ and `efConstruction`$= 500$.
- **PAG** (Lu et al., 2026) augments graph construction with projection information. We use its official implementation with $M = 32$ and a construction beam width of 100.

Table 2: Datasets used in the main evaluation.

Dataset	#Base vectors	#Queries	Dim	Explanation
Cohere	1,000,000	10,000	768	Cohere Wikipedia embeddings
DINOv2	1,281,167	10,000	768	ImageNet DINOv2 features
FB-SSNPP	1,000,000	1,000	256	SimSearchNet++ product embeddings
Gaussian	1,000,000	1,000	256	Independent standard Gaussian vectors
GloVe-200	1,193,514	1,000	200	Twitter 27B GloVe embeddings
GloVe-300	2,196,017	1,000	300	Common Crawl 840B GloVe embeddings
ImageNet	2,340,373	200	150	ImageNet descriptors
SproosV3	1,000,000	10,000	4,096	Sproos v3-large neural activations
Tiny5M	5,000,000	1,000	384	GIST descriptors from Tiny Images
WikiGiga	1,000,000	10,000	300	Wikipedia and Gigaword GloVe embeddings
Word2Vec	1,000,000	1,000	300	Google News Word2Vec embeddings
ZZSN	1,000,000	10,000	1,280	UNet attention activations

Table 3: Selector stability across estimator settings and random seeds. The reported values of $\widehat{\Delta}_e$, $\widehat{\Delta}_s$, and $\widehat{\Phi}_S$ use $L_e = 64$ and $B_H = 8$. The "All settings" column covers 135 configurations from three values of L_e, three values of B_H, and fifteen seed configurations. The "$L_e = 64$, $B_H = 8$" column varies the fifteen seed configurations at those settings.

Dataset	Best	$\widehat{\Delta}_e$	$\widehat{\Delta}_s$	$\widehat{\Phi}_S$	All settings (%)	$L_e = 64$, $B_H = 8$ (%)
Cohere	ASC	-1.000	-0.050	-4.551	100.0	100.0
DINOv2	RAND	-0.969	0.098	0.142	100.0	100.0
FB-SSNPP	ASC	-1.000	-0.002	-3.067	100.0	100.0
Gaussian	DESC	1.000	0.000	-3.013	100.0	100.0
GloVe-200	DESC	0.656	-0.008	-2.231	100.0	100.0
GloVe-300	DESC	0.500	-0.089	-4.262	97.8	100.0
ImageNet	ASC	-1.000	-0.058	-4.792	100.0	100.0
SproosV3	ASC	-1.000	0.026	-2.192	100.0	100.0
Tiny5M	ASC	-1.000	0.008	-2.757	100.0	100.0
WikiGiga	DESC	0.969	-0.031	-3.858	100.0	100.0
Word2Vec	ASC	-1.000	-0.010	-3.308	100.0	100.0
ZZSN	RAND	-1.000	0.237	4.349	100.0	100.0

- **Möbius-Graph** (Zhou et al., 2019) converts MIPS into a metric-search problem through a norm transformation. We use the official implementation and its default transformation exponent.
- **ScaNN** (Guo et al., 2020) combines partitioning, anisotropic vector quantization, and exact reordering. We use dot-product search with 2,000 leaves and 100,000 training vectors.
- **SOAR** (Sun et al., 2023) improves ScaNN's partitioning by assigning each vector to two centers. We use the official SOAR configuration with the same ScaNN backbone.
- **MAG** (Chen et al., 2025a) combines inner-product and Euclidean neighborhoods in a graph index. We use the authors' configuration on a 400-neighbor NN-Descent graph (Dong et al., 2011).
- **PSP** (Chen et al., 2025b) constructs a proximity graph from query-scaled nearest-neighbor geometry. It uses the same 400-neighbor NN-Descent graph as MAG and the authors' recommended parameters.

B.2 Selector Stability and Constant Sensitivity

We evaluate selector stability at $S = 32$. We first fix $L_e = 64$ and $B_H = 8$ and test fifteen configurations formed from five direction seeds and three independent groups of eight hyperplane sequences. We then repeat these configurations for $L_e \in \{64, 128, 192\}$ and $B_H \in \{4, 8, 16\}$, giving 135 configurations per dataset. Table 3 reports the coefficient values at $L_e = 64$ and $B_H = 8$,

Table 4: Selection accuracy and throughput regret at $k = 10$ as the graph-degree parameter M changes. Regret is the geometric-mean ratio between the best measured matched-recall throughput and a strategy's throughput, minus one. Best fixed is the single insertion order with the lowest regret at each M.

M	Prediction accuracy	Selector regret	Best fixed	Fixed regret	RAND regret
16	79.2%	3.83%	ASC	44.67%	49.66%
32	100.0%	0.00%	ASC	42.12%	57.17%
64	95.8%	1.58%	ASC	35.85%	66.63%

Table 5: Sensitivity to the target occupancy used to set the angular resolution. Each entry reports the predicted order after recomputing b_S and $\widehat{\Delta}_s$. The final column gives b_S for the three targets.

Dataset	Best	Target S	Target $2S$	Target $4S$	b_S for targets $S/2S/4S$
Cohere	ASC	ASC	ASC	ASC	33/30/29
DINOv2	RAND	RAND	RAND	ASC	16/14/13
FB-SSNPP	ASC	ASC	ASC	ASC	114/109/104
Gaussian	DESC	DESC	DESC	DESC	15/14/13
GloVe-200	DESC	DESC	DESC	DESC	16/15/14
GloVe-300	DESC	DESC	DESC	DESC	19/18/16
ImageNet	ASC	ASC	ASC	ASC	24/21/19
SproosV3	ASC	ASC	ASC	ASC	26/22/19
Tiny5M	ASC	ASC	ASC	ASC	50/46/44
WikiGiga	DESC	DESC	DESC	DESC	16/15/14
Word2Vec	ASC	ASC	ASC	ASC	18/16/15
ZZSN	RAND	RAND	RAND	RAND	15/14/13

the accuracy over all 135 configurations, and the accuracy over the fifteen seed configurations at $L_e = 64$ and $B_H = 8$.

All fifteen seed configurations at $L_e = 64$ and $B_H = 8$ select the measured best order on every dataset. Across all 135 configurations, eleven datasets are correct in every case. GloVe-300 is correct in 132 configurations, with all three errors occurring at $B_H = 4$ and none at $B_H \geq 8$. DINOv2 and ZZSN select RAND in all 135 configurations. The 2.5th and 97.5th percentiles of $\widehat{\Phi}_S$ are 0.09 and 0.77 for DINOv2 and 2.73 and 3.95 for ZZSN. The reported ZZSN value is 4.35 and uses a separate group of eight hyperplane sequences from the main experiments. Every one of the 135 ZZSN margins remains positive.

We further examine the constant in the decision rule by replacing 3 with λ and evaluating $(S - 1)\widehat{\Delta}_s > \lambda|\widehat{\Delta}_e|$. Using the reported coefficient values, all twelve decisions remain correct for $\lambda \in [0.808, 3.147)$, with SproosV3 and DINOv2 determining the lower and upper endpoints. Across the fifteen seed configurations at $L_e = 64$ and $B_H = 8$, the common interval is $\lambda \in [0.711, 3.085)$, with ImageNet and DINOv2 determining the two endpoints. The value $\lambda = 3$ used in equation 12 lies within both intervals.

B.3 CAPACITY AND ANGULAR-SCALE SENSITIVITY

We first examine whether the selector remains effective when the graph degree changes. We test $M \in \{16, 32, 64\}$ and set $S = M$ in each experiment. For every combination of dataset, graph builder, and degree, we rebuild the graphs under ASC, DESC, and RAND and evaluate them at $k = 10$. This gives 24 conditions at each degree.

To compare two strategies, we restrict their QPS–recall curves to the recall interval shared by all three orders. We interpolate log-QPS at 101 uniformly spaced recall values and use the geometric-mean QPS over these values as the score. The multiplicative regret of a strategy is the geometric mean, over the 24 conditions, of the ratio between the best measured score and the strategy's score, minus one. Thus, a regret of 4% means that the best measured order is $1.04\times$ faster on average.

Table 6: Later-position pruning under DESC and ASC. The second column reports the measured best insertion order. Retained degree is the number of neighbors remaining after pruning, rejection rate is the fraction of tested candidates discarded by a retained neighbor, and blocker/source norm ratio is the average r_z/r_x in the pruning condition equation 1. The final three columns report DESC − ASC, with positive values indicating a larger statistic under DESC.

Dataset	Measured best	Retained degree	Rejection rate	Blocker/source norm ratio
Cohere	ASC	−8.245	0.067	0.070
DINOv2	RAND	−0.938	0.016	0.031
FB-SSNPP	ASC	−30.599	0.308	0.040
Gaussian	DESC	0.000	0.144	0.078
GloVe-200	DESC	−4.748	0.214	0.375
GloVe-300	DESC	−6.509	0.124	0.457
ImageNet	ASC	−22.726	0.099	1.524
SproosV3	ASC	−10.689	0.031	0.269
Tiny5M	ASC	−25.095	0.125	0.928
WikiGiga	DESC	−5.611	0.195	0.497
Word2Vec	ASC	−23.781	0.121	4.498
ZZSN	RAND	−6.078	0.024	0.446

Table 4 summarizes the results. The selector agrees with the measured winner in 66 of the 72 conditions and has less than 4% multiplicative regret at every degree. It is correct in 19 of 24 conditions at $M = 16$, all 24 conditions at $M = 32$, and 23 of 24 conditions at $M = 64$. Two of the five disagreements at $M = 16$ are very close: the selected order achieves 96.7% and 91.6% of the best matched-recall throughput on Cohere and ZZSN with PAG, respectively. Among the three fixed strategies, ASC performs best at every degree, but its regret ranges from 35.85% to 44.67% and is consistently larger than that of the selector.

We next test the sensitivity of $\widehat{\Delta}_s$ to the angular resolution. Equation 8 sets b_S by requiring each vector to have at least $2S$ other vectors in the same region on average, matching the S retained candidates and S alternatives in the exchange model. We repeat the calculation using targets S and $4S$, which produce finer and coarser partitions, respectively. As shown in Table 5, changing the target from $2S$ to S leaves all twelve selections unchanged. With the coarser $4S$ target, eleven selections remain unchanged, while DINOv2 moves from RAND to ASC. Thus, the selector is stable across this fourfold range of angular scales, with DINOv2 as the only exception.

B.4 LATER-POSITION PRUNING DIAGNOSTICS

The pruning condition in Section 3.1 depends on the norm ratio between the blocker and the source. We therefore test whether this asymmetry in later pruning can itself predict the preferred insertion order. For each dataset, we compare three quantities under DESC and ASC: the degree retained after pruning, the fraction of candidates rejected by retained neighbors, and the average blocker-to-source norm ratio. Table 6 reports their differences, computed as DESC − ASC and averaged over eight construction runs.

Compared with ASC, DESC consistently uses higher-norm blockers and rejects a larger fraction of candidates. It also produces smaller retained lists on eleven datasets, with Gaussian as the only tie. These signs are shared by datasets favoring ASC, DESC, and RAND. The magnitude of the retained-degree reduction is more informative: it is at least 8.2 neighbors on every ASC-preferred dataset and at most 6.5 on every DESC-preferred dataset. This is consistent with DESC being harmful when it leaves neighbor lists substantially underfilled, as in Figure 1. These construction-time statistics show that the two sorted orders differ substantially after the first position. The utility averages these later-position results and uses Δ_e to compare ASC with DESC before construction.

B.5 COMPARISON WITH SIMPLE GEOMETRIC STATISTICS

We compare the selector with six one-dimensional summaries of the base vectors. For each dataset, we sample $100{,}000$ vectors uniformly without replacement and write $\bar{r}$ and s_r for the sample mean

Table 7: Accuracy of simple one-dimensional dataset statistics. The oracle chooses the best three-interval rule after observing all twelve outcomes. Leave-one-out uses the nearest scalar value among the other eleven datasets.

Statistic	Definition	Oracle	Leave-one-out
Norm variation	$s_r/\bar{r}$	$9/12$ (75.0%)	$4/12$ (33.3%)
Mean-direction anisotropy	$\|\mathbb{E}_i[\boldsymbol{u}_i]\|_2^2$	$8/12$ (66.7%)	$5/12$ (41.7%)
Pair angular concentration	$\mathbb{E}_{i\neq j}[(\boldsymbol{u}_i^\top \boldsymbol{u}_j)^2]$	$9/12$ (75.0%)	$3/12$ (25.0%)
Directional participation ratio	Reciprocal of pair angular concentration	$9/12$ (75.0%)	$4/12$ (33.3%)
Norm–direction coupling	$\|\mathbb{E}_i[(r_i-\bar{r})\boldsymbol{u}_i/s_r]\|_2^2$	$8/12$ (66.7%)	$4/12$ (33.3%)
Top–bottom directional shift	Squared distance between the mean directions of the largest- and smallest-norm deciles	$7/12$ (58.3%)	$0/12$ (0.0%)

Table 8: Accuracy of direct insertion-order decision rules. Here, "sorted" denotes the ten datasets whose measured winner is ASC or DESC.

Rule	Decision	Accuracy
Always ASC	ASC	$6/12$ ($6/10$ sorted)
Global pair threshold	DESC if $\bar{\gamma} < 1/S$, else ASC	$7/12$ ($7/10$ sorted)
Using $\widehat{\Delta}_e$	DESC if $\widehat{\Delta}_e > 0$, else ASC	$10/12$ ($10/10$ sorted)
Using both coefficients	RAND if $\widehat{\Phi}_S > 0$, otherwise use $\widehat{\Delta}_e$	$12/12$

and standard deviation of the norms. Pairwise quantities are estimated from $131{,}072$ distinct pairs drawn from a reservoir of $8{,}192$ directions sampled from the same $100{,}000$ vectors.

We convert each scalar into a three-way rule with an in-sample oracle that considers every placement of two thresholds and every assignment of the resulting intervals to ASC, DESC, and RAND. The thresholds may coincide or lie outside the observed range, so an interval may be empty. We also report leave-one-out 1-NN accuracy, assigning each dataset the measured preference of the dataset with the closest scalar value among the remaining eleven.

Table 7 shows that no scalar statistic exceeds $9/12$ even after its thresholds and interval labels are chosen from all twelve outcomes, and the highest leave-one-out accuracy is $5/12$. Because the oracle permits empty intervals and arbitrary label assignments, any scalar that separated the six ASC datasets from the four DESC datasets with one threshold would attain at least $10/12$ by leaving the RAND interval empty. The observed ceiling of $9/12$ therefore shows that none of these scalars alone separates the two norm-sorted preferences. The participation ratio is a monotone transformation of pair angular concentration and therefore has the same oracle accuracy. Its leave-one-out accuracy differs because nearest neighbors depend on the numerical scale of the statistic.

We next compare four rules that directly output an insertion order. Write $\bar{\gamma} = \mathbb{E}_{i\neq j}[(\boldsymbol{u}_i^\top \boldsymbol{u}_j)^2]$ for the global pair concentration. The rule based on $\widehat{\Delta}_e$ applies the threshold to the top-$2S$ candidates along each sampled direction. The rule using both coefficients also uses $\widehat{\Delta}_s$. The global pair threshold makes errors in both directions. SproosV3 and Word2Vec have $\bar{\gamma} < 1/S$ but favor ASC, whereas WikiGiga has $\bar{\gamma} \geq 1/S$ but favors DESC. The value $\widehat{\Delta}_e$ instead evaluates redundancy within the top-$2S$ candidates along each sampled direction and selects all ten norm-sorted datasets. These two rules differ both in their candidate sets and in how they aggregate directions, so their comparison indicates the benefit of evaluating redundancy locally. The rule using both coefficients then uses $\widehat{\Delta}_s$ to identify the two datasets that favor RAND.

B.6 RESULTS FOR TOP-100 RETRIEVAL

Figure 4 extends the insertion-order comparison to $k = 100$. The preferred order remains unchanged: six datasets favor ASC, four favor DESC, and DINOv2 and ZZSN favor RAND. The same pattern holds for both ip-NSW and PAG, showing that the preferred insertion order is stable across graph builders and retrieval depths. The distinct QPS–recall curves further confirm that insertion order continues to affect search performance when each query retrieves a larger result set.

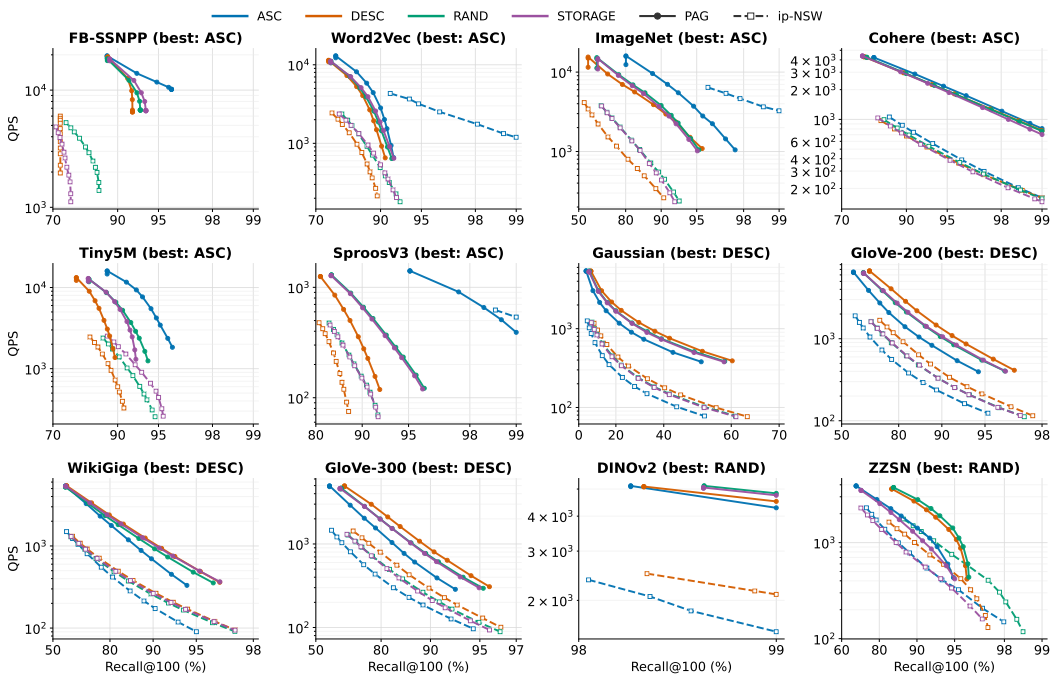


Figure 4: QPS–recall tradeoffs at $k = 100$ for ip-NSW and PAG under ASC, DESC, RAND, and the original storage order. Construction and search parameters are fixed within each method. The RAND curve averages five independently sampled insertion orders, while the storage-order curve uses the vector order in the released base file.

B.7 Comparison with MIPS Baselines

Figures 5 and 6 compare the QPS–recall curves of ip-NSW and PAG, using the insertion order selected by our rule, with five MIPS baselines: Möbius-Graph, ScaNN, SOAR, MAG, and PSP. Each point represents a measured search configuration; at the same recall, a higher curve indicates greater query throughput. At both $k = 10$ and $k = 100$, the selected-order ip-NSW and PAG curves form or closely follow the best measured frontier on most of the twelve datasets, with particularly clear advantages in the high-recall region of the embedding and neural-activation datasets. The relative behavior is also consistent across the two retrieval depths, so the competitive frontier extends across both values of k. These results demonstrate the practical value of insertion-order selection: it allows existing incremental graph builders to preserve and, on many datasets, extend their performance advantage over dedicated MIPS baselines.

B.8 Comparison with Subsample Trial Construction

A natural alternative is to construct the three insertion orders on a small subset and select the order with the best QPS–recall curve on that subset. We evaluate this baseline at $k = 10$ using either a uniform 5% sample or the 5% of vectors with the largest norms. The three orders are constructed and evaluated separately for ip-NSW and PAG using the same parameter settings as the full-data experiments. We report the experimental results as follows.

For ip-NSW, both sampling strategies recover the full-data winner on 11 of 12 datasets. Uniform sampling selects DESC instead of ASC on FB-SSNPP, while largest-norm sampling selects ASC instead of RAND on DINOv2. For PAG, both strategies recover the winner on 9 of 12 datasets. Uniform sampling makes different selections on DINOv2, FB-SSNPP, and GloVe-200, while largest-norm sampling makes different selections on Cohere, DINOv2, and FB-SSNPP. Across the four sampling conditions, trial construction recovers the full-data winner in 40 of 48 cases, leaving eight different selections. In addition, this baseline requires constructing three graphs, measuring three

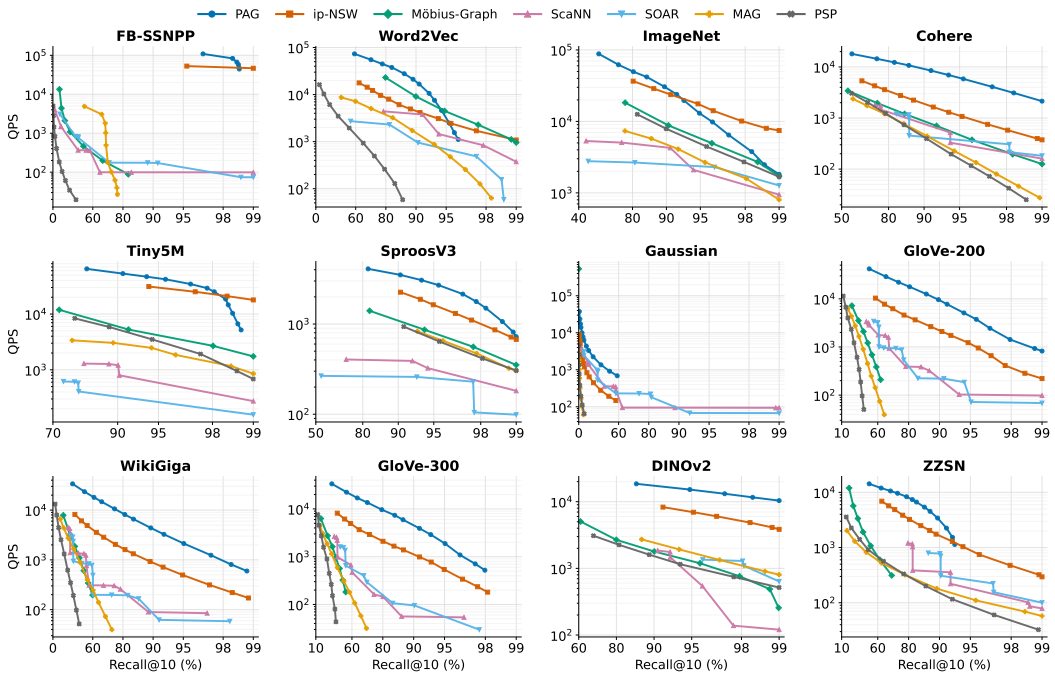


Figure 5: QPS–recall comparison with MIPS baselines at $k = 10$. For ip-NSW and PAG, each curve uses the insertion order predicted by our selector from among ASC, DESC, and RAND.

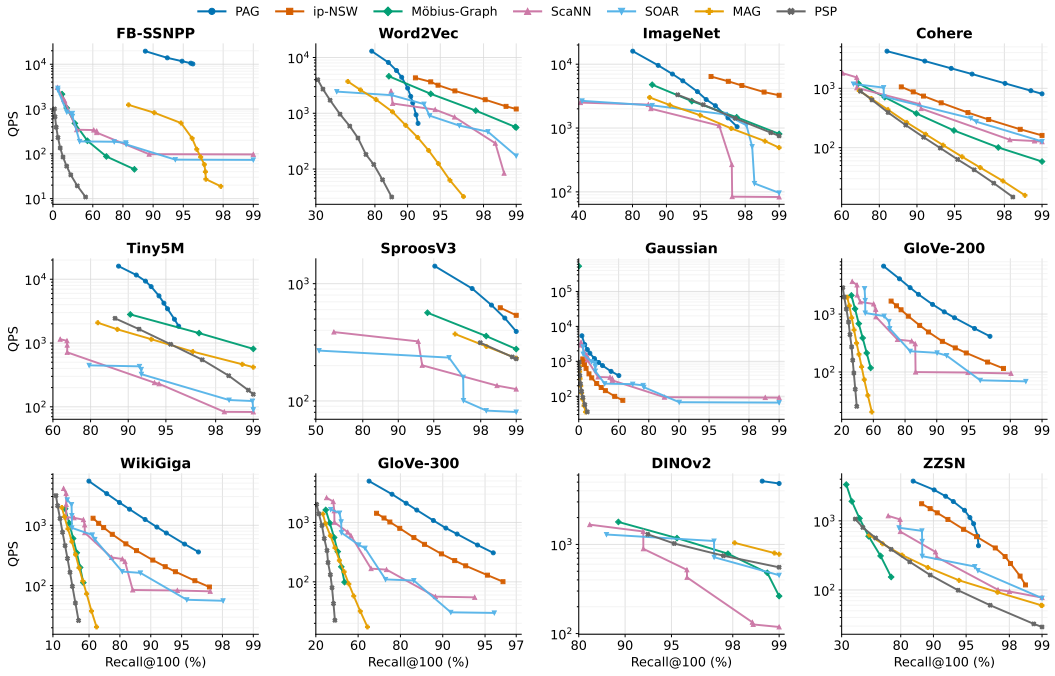


Figure 6: QPS–recall comparison with MIPS baselines at $k = 100$. For ip-NSW and PAG, each curve uses the insertion order predicted by our selector from among ASC, DESC, and RAND.

complete QPS–recall curves, and comparing them over a common recall range, making the selection procedure considerably more involved.