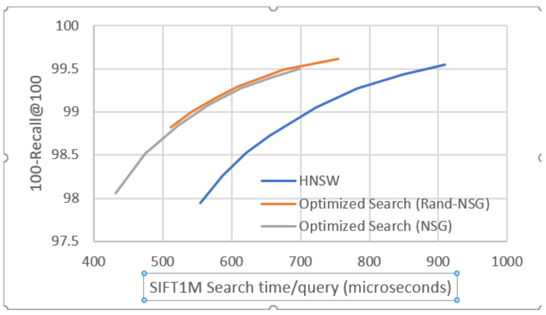
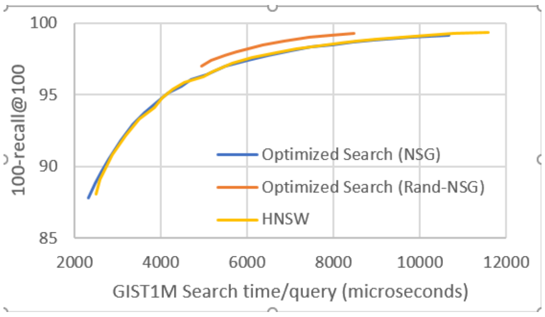
Rebuttal for submission 7667 - Rand-NSG

Reviewer 1: We will release our code along with the paper.

Reviewer 2:
Our main goal was to enable high performance SSD-friendly indices on inexpensive workstations. Towards this goal, as you have suggested, we have adapted known techniques, and, where necessary, contributed new techniques as well. A few contributions are: (1) We introduce a new pruning strategy parameterised by α that reduces the number of hops from navigating vertex to any other vertex. This provides a greater control over the diameter of the graph which is important for disk search. This is a feature that previous methods like NSG and HNSW lack. Using our pruning with a higher $\alpha > 1$ parameter will yield more longe range edges and reduces the diameter of the graph. The practical benefits of this idea can be seen in Figure 2(b) in the submission, where we notice that the number of hops required for search improved with the degree of the graph for Rand-NSG, and plateaus for NSG and HNSW. (2) We also introduce a 2-pass construction algorithm, that allows us to use lower L for index construction, thus improving the construction speed necessary for a target graph index quality (compared to NSG and HNSW). (3) We provide a high quality implementation of disk-based search.





QPS plots: We will expand on the plots above comparing NSG, HNSW, and Rand-NSG. Rand-NSG indices can be searched with NSG in-memory search (we did not implement a separate in-memory search since our focus was on disk search). The plots above suggest that Rand-NSG indices are competitive with NSG and HNSW. Considering that they use roughly the same search algorithm for querying, we focused on the number of distance comparison per query as a machine-agnostic way of comparing the algorithms.

Reviewer 3: Thank you for pointing out citations we have missed or erred on (e.g., in lines 27, 127, 154, 196, 249). We will update related work, give due credit to missed citations and do a pass over the writing. Re. other questions:

- Yes, base points mean domain points. We'll clarify the notation.

22-23 : Why not $200 - 300$? Could well be, we have experimented with 960-dimensional GIST in the paper.

29 : We will define the recall formally in the final version. *a-recall@b* means the percentage of points among the a points that intersect with the true b nearest neighbor. We'll also clarify that the goal is maximizing search efficiency for a given recall, rather than the recall itself.

37 : We will temper the claim and suggest that graph-based methods produce some of the best indices. We'll give NN-descent algorithms credit in related work.

39 : We do not have a concrete definition of *navigable graphs*; we roughly mean those graphs on which greedy-like iterative search heuristics converge rapidly.

40 : We will clarify the differences between our search and pure greedy search.

- We will report numbers on Deep1B. We have preliminary evidence suggesting that we can achieve 95% recall@1 with 5ms latency and 30GB working memory.

We'll also discuss the pruning strategy used by HNSW and NSG in our final version, and contrast with Rand-NSG.