In the above figures, we show that FreshVamana ensures stable search recall for any recall target. In our paper, we had provided evidence for recall stability for 5-recall@5. Here, our experiments show that even 1-recall@1, 10-recall@10 and 100-recall@100 are stable.

Additional Data: Figures for k-recall@k vs latency

In each of the following 4 figures, we compare recall vs latency for static and an equivalent incremental index on different datasets. We use build parameters R as 64 and L as 75 for building the static index. An equivalent incremental index means one that is built with build parameters adjusted such that it takes the same time to build the index as a static index (in our paper we have explained about why an incremental index build using the same parameters as a static index would be much faster). In each figure, we plot 1-recall@1, 10-recall@10 and 100-recall@100 for both the indices.

1. Vamana static index and incremental index on SIFT1M.
2. Vamana static index and incremental index on DEEP1M.
3. Vamana static index and incremental index on GIST.
4. Vamana static index and incremental index on SIFT100M.