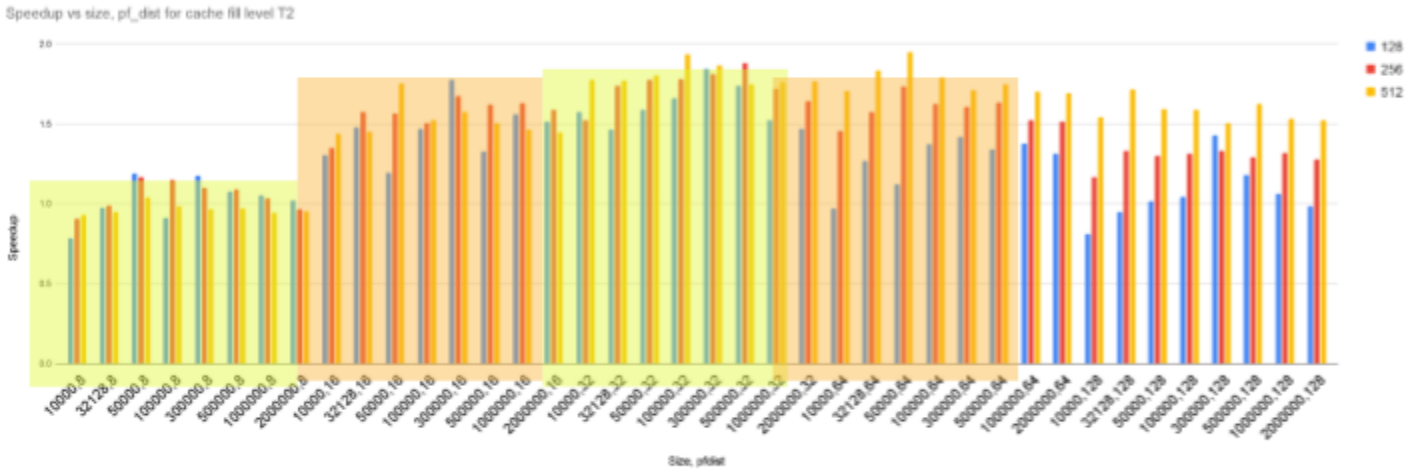
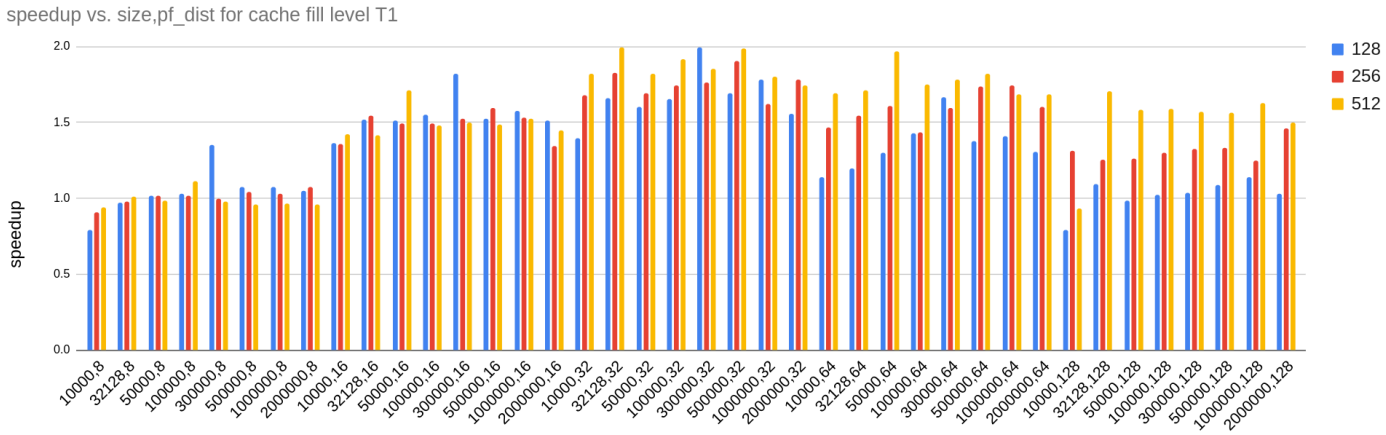
Task 2A: Software prefetching

1. *Analyze the Impact of Embedding Table Size*:

- Increase the size of the embedding table and observe how software prefetching performance changes.



similarly,



Software prefetching performance shows minimal changes as sizes are changed over different prefetch distances.

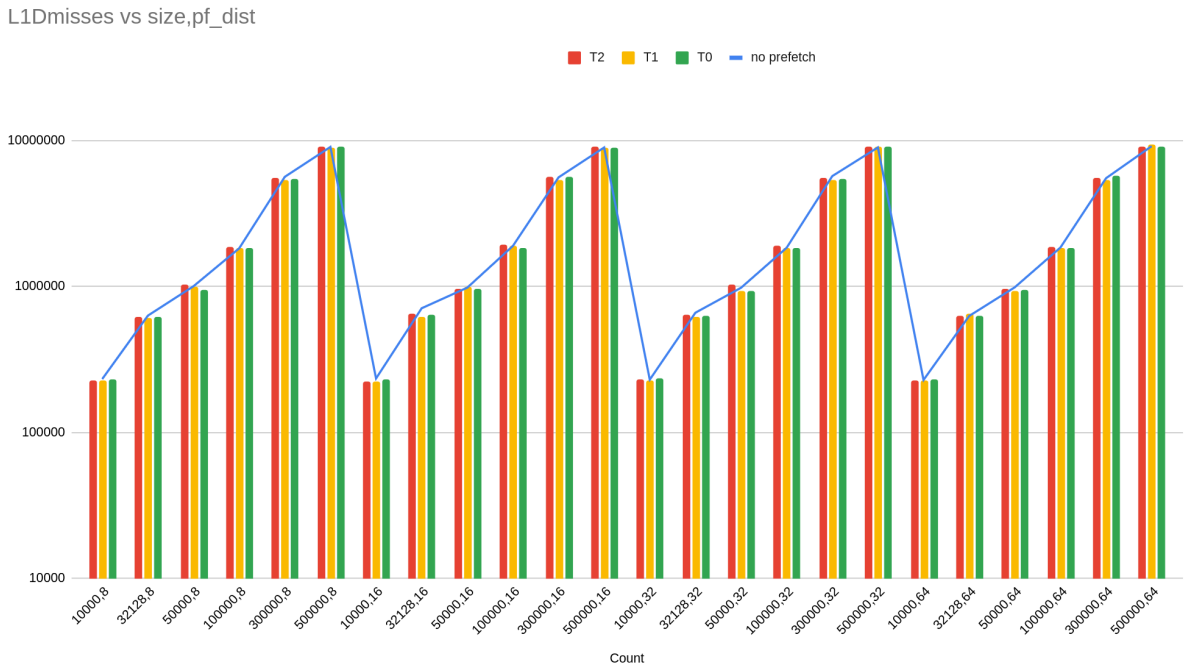
2. *Analyze CPU Metrics*:

For different embedding table sizes and different software prefetching parameters, analyze the trends in key CPU metrics (use the perf tool):

- *L1D Cache Misses*
- *L2 Cache Misses*
- *LLC (Last Level Cache) Misses*

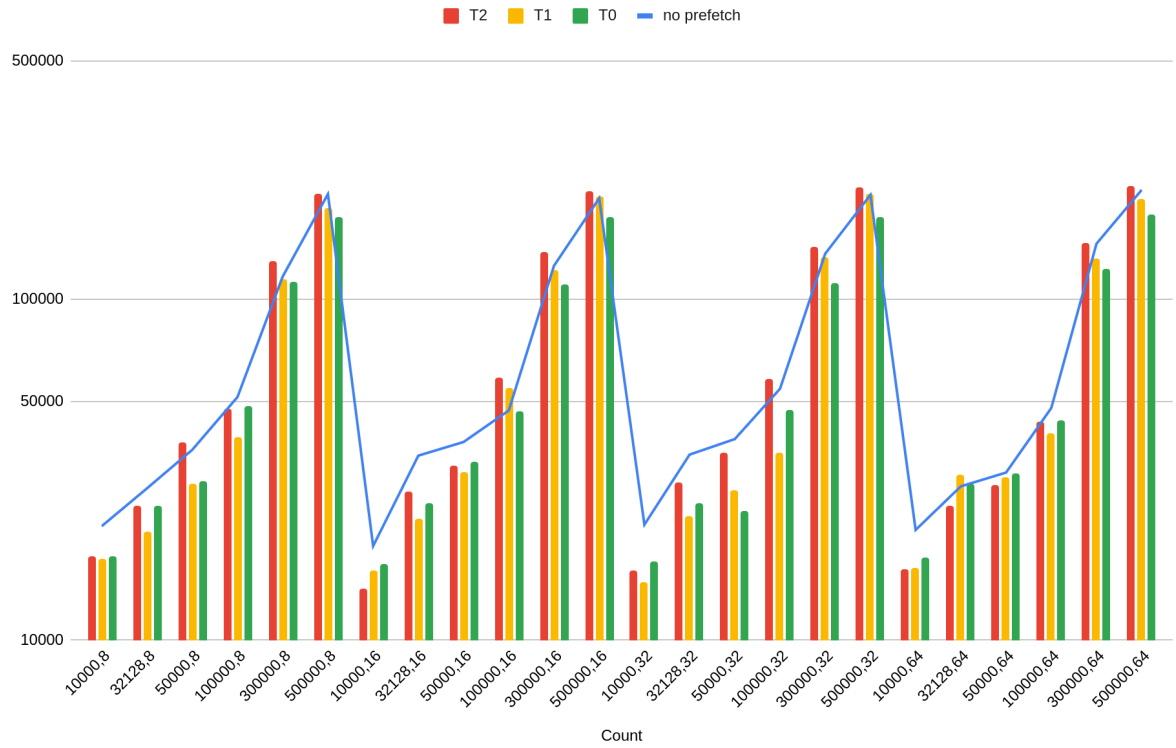
Ans: There wasn't much difference in the number of misses in L1D caches by implementing software prefetching at different cache fill levels (T0, T1, and T2). However we observed a reduction of misses in LLC and L2 at cache fill level T1 and T0.

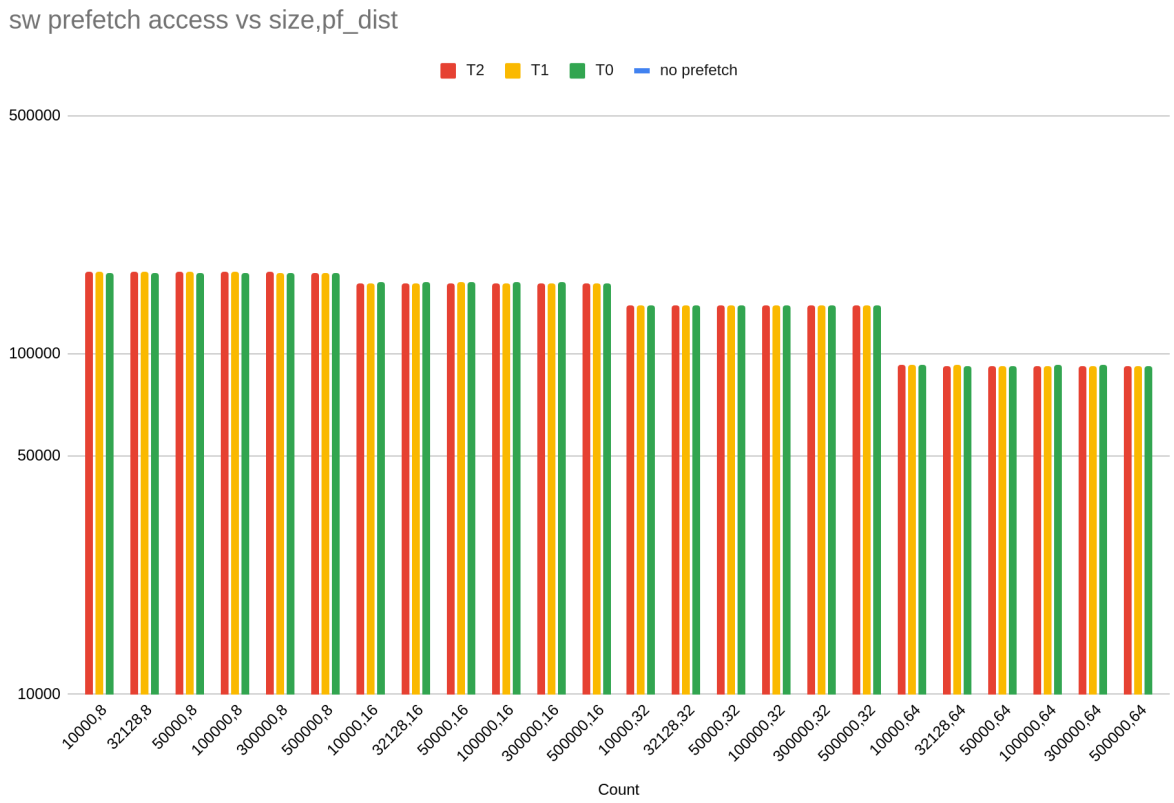
Note : The following charts are made for dimension size 128.



L2 misses vs size,pf_dist

LLC misses vs size,pf_dist





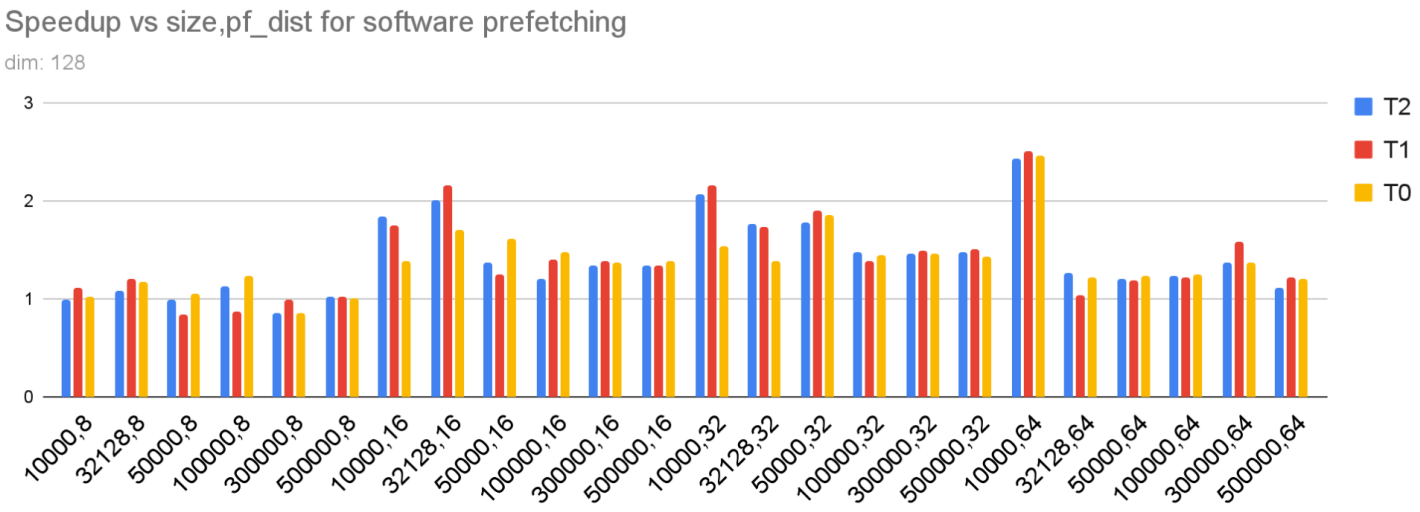
For similar data for dimension sizes 256 and 512, refer this [sheet](#).

3. *Collect Execution Time and Compute Speedup*:

- Measure the execution time of the embedding operation with and without software prefetching.
- Compute the speedup as follows:

$$Speedup = \frac{Execution\ Time\ without\ Software\ Prefetching}{Execution\ Time\ with\ Software\ Prefetching}$$

- Analyze how the speedup varies with different embedding table sizes and different software prefetching parameters.



4. *Identify Optimal Parameters*:

Determine the optimal prefetch distance and optimal cache fill level based on your observations.

Ans: Optimal Prefetch distance: **32;** Optimal cache fill level: **T1.**

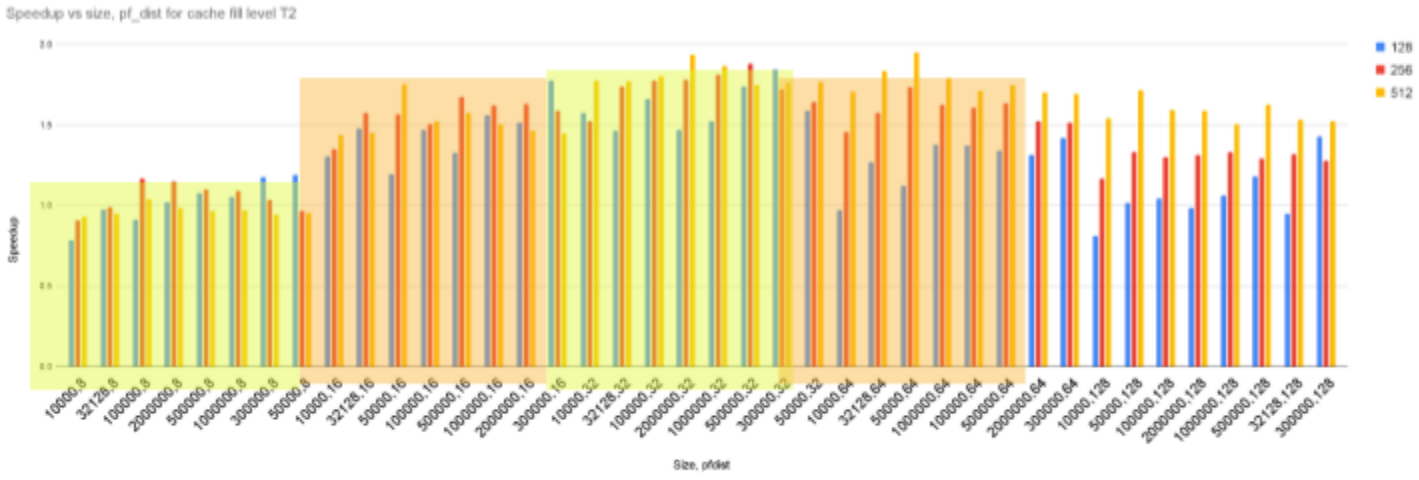
5. *Study the Effect of Hardware Prefetchers*:

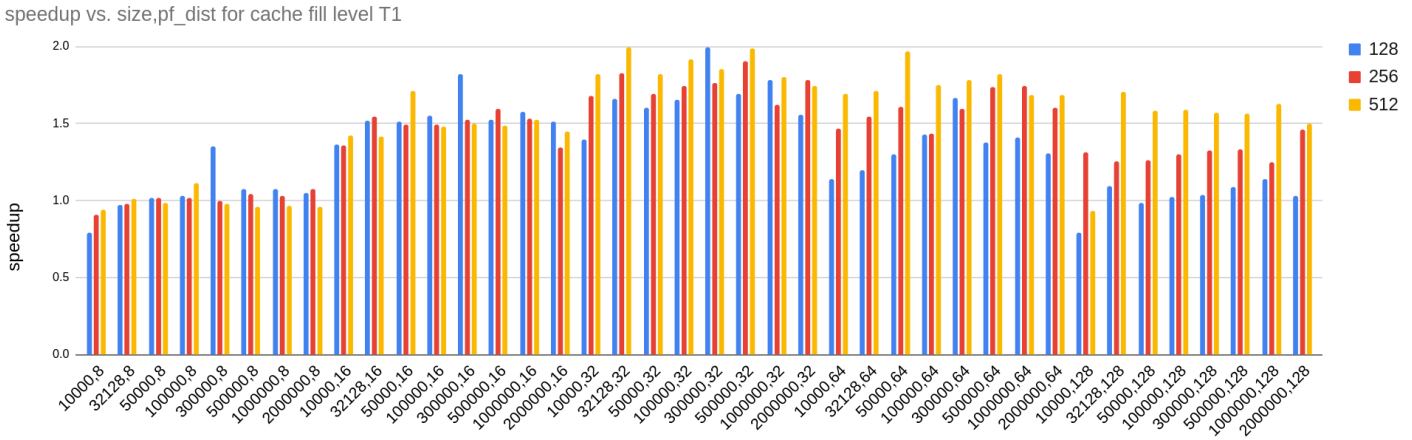
Enable and disable hardware prefetchers and analyze their impact on the embedding operation's performance.

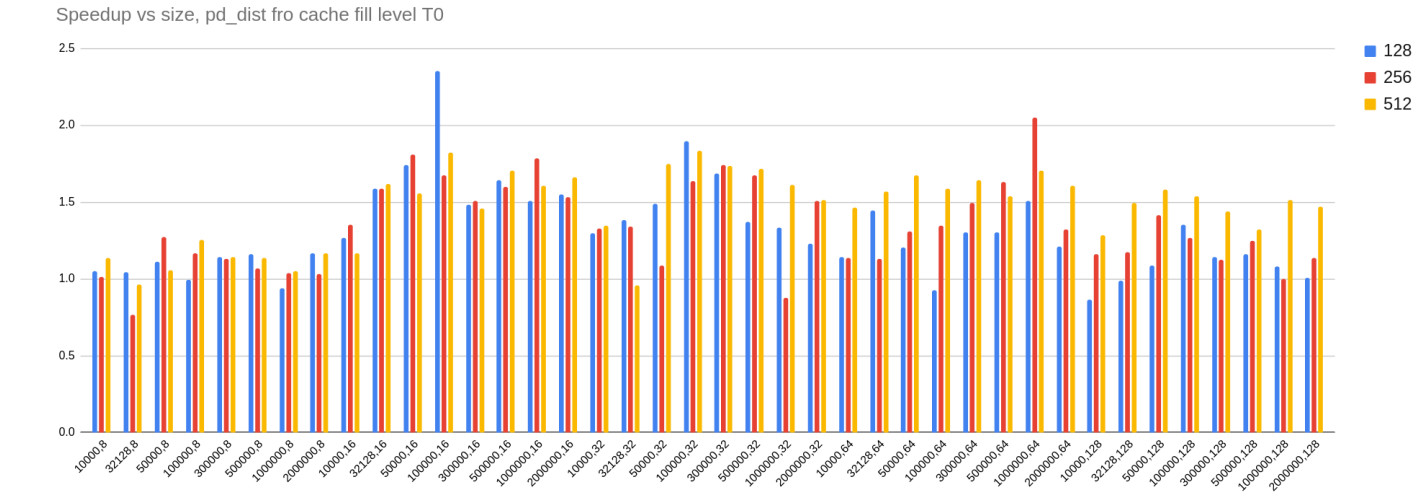
Ans: When hardware prefetching was enabled, we observed a degradation in performance while implementing software prefetching on all cache fill levels and all prefetch distances.

1. *What trend do you observe in speedup with different embedding table sizes?*

Ans: The speedup did not change much with increasing sizes, except for a slight dip for larger sizes in some cases for cache fill level T1 and T2. For T0, there was an a peak towards size 100000 where prefetch distance was 64 or 32.







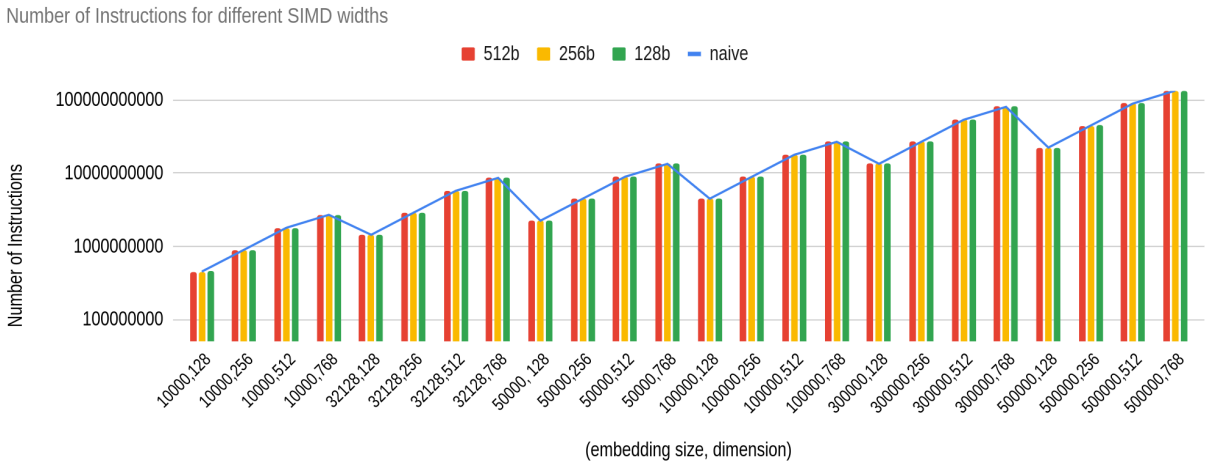
2. *What is the best prefetch distance?*

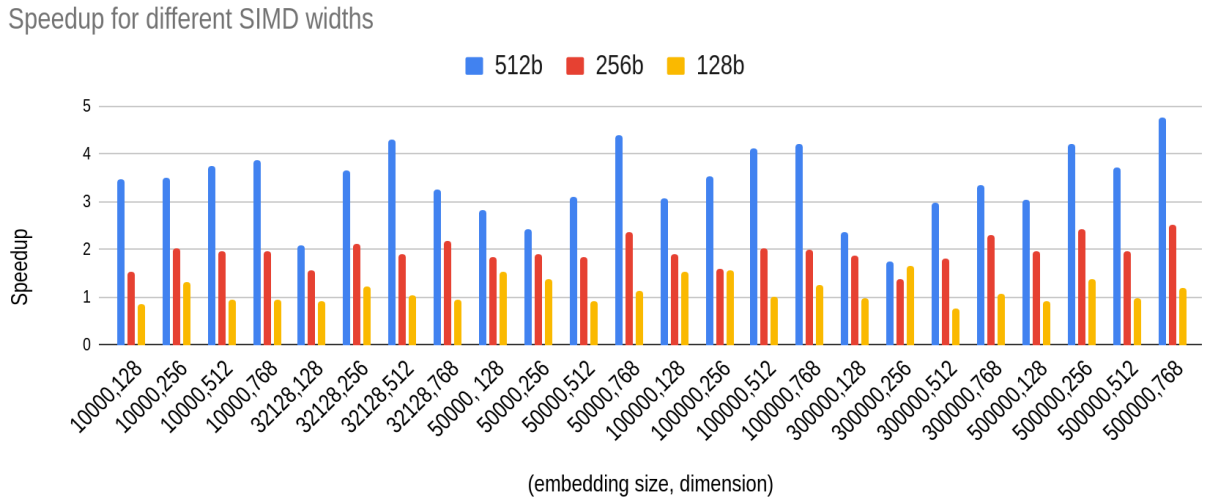
Ans: We concluded that 32 was the best prefetch distance for all cache fill levels and dimensions, although 64 gave a comparable performance.

3. *At what cache fill level do you achieve the maximum speedup*

Ans: We saw the best speedup at T1 cache fill level.

Task 2B: SIMD (Single Instructor Multiple Data)





1. *What trends do you observe in speedup for different combinations of embedding dimensions and SIMD widths?*

Ans: There was a rough trend of increasing speedup with increasing dimension sizes across all embedding table sizes.

2. *For which SIMD width do you achieve the maximum speedup?*

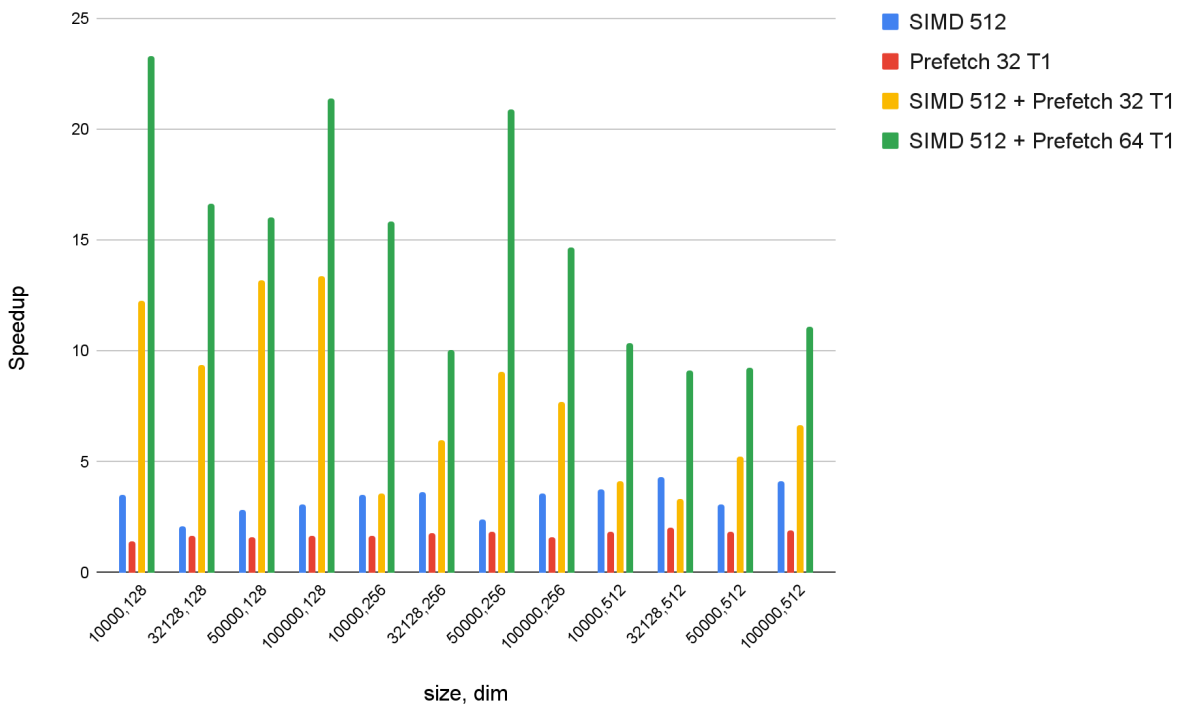
Ans: **512b** SIMD gave the best speedup.

Task 2C: Software prefetching + SIMD

Final performance summary for Embedding Operation:

Include a plot that compares the best performance achieved by each optimization technique —

- Software Prefetching
- SIMD
- Software Prefetching + SIMD



For more comparisons, refer to tables on this [sheet](#).

We can conclude that combining simd efficiently with prefetching can drastically improve performance.