# 1 Response to Reviewer 1

**Re assumption 1:** Shifting the data points is a good idea, but it might cause problems. Shifting the data points changes the norms of all vectors, while the norms are very important quantities in the MIPS problem. Without shifting, the greedy search algorithm exploits from well-chosen top-norm vectors, but this advantage is no longer valid after shifting.

In our current work, we focus on theory and datasets satisfying assumption 1. Our approach shows improvement in these datasets. We have not implemented our algorithm on datasets violating assumption 1, so we did not make a conclusion about shifting. Indeed, we have not found any datasets violating assumption 1 in recommendation system, but we will further explore related experiments in artificial datasets in our future work. Further discussions about assumption 1 and shifting appear in Appendix C in the supplemental file.

**Re normal assumption:** Thanks for pointing this out. Normal distribution assumption is much more than enough and might not be realistic. We will rephrase the sentence as follows: "In these scenarios, the entries of data vectors distributed on the whole real line. With high probability, each hyperoctant contains at least one data point so that the convex hull of the dataset contains 0 as an interior point."

**Re GPU acceleration:** Thanks for your suggestions on the relevant literature. We will definitely add them to our reference and further discuss their work in the section of related work. Indeed, GPU+exact search performs well on some datasets, but we believe approximate search methods have the advantage in scalability on larger datasets.

In the present work, we aim to improve the efficiency of the MIPS problem in algorithmic perspective. However, the implementation of our method on GPU-platform can be an interesting and novel extension. GPU can process multiple queries in parallel. GPU streaming multiprocessors can compute inner products very fast. Nevertheless, in our approach, the set of priority queue (set $C$ in Algorithm 1) and the indices of visited vertices can be arbitrarily large. Theoretically, their sizes only have a trivial upper bound, the number of total vertices $n$. It does not hurt the efficiency for implementation on CPU, but straightforwardly transplanting this algorithm to the GPU is problematic. We will add extra discussion in the paper and leave the details for future work.

**Re preprocessing time:** In our experiments, preprocessing time only includes graph construction time, which is presented in Table 2, at the end of Section 5.3, in line 281. We compare the graph construction time with ip-NSW. As can be seen in the table, our approach consumes much less time during the preprocessing procedure. The table is duplicated as follows:

Table 2 Graph Construction Time in Seconds.

| | Netflix | Amovie | Yelp | Music-100 |
|---|---|---|---|---|
| ip-NSW | 2.19s | 36.95s | 6.78s | 396.82s |
| Möbius-Graph | 1.89s(-13.7%) | 24.35s(-34.1%) | 2.34s(-65.5%) | 162.24s(-59.1%) |

**Re contribution:** Thanks for finding out our work is interesting. We agree our work is based on existing approximate nearest neighbor (ANN) search on graph algorithms, which have been studied for about 15 years. It has been improving and applying to various search tasks. It was shown that well-designed heuristics can significantly improve searching performance for ANN search, but generalization of these ideas to the MIPS is non-trivial. The goal of our paper is to fill this gap.

# 2 Response to Reviewer 2

Thank you so much for highly encouraging comments. We will change the sizes of markers in Figure 2 and Figure 3. We address your concern about the normal assumption in our response to reviewer 1. The normal assumption is indeed not necessary. We will rephrase the sentence more accurately.

# 3 Response to Reviewer 3

We appreciate your detailed nice summary of our work. We will move some plots from Figure 5 and 6 to Section 2 if space allows. We will also change "Möbius" to "Möbius-graph" in Figure 5, 6 and 7.