A Appendix

A.1 Experiment Configurations

To obtain our REFuSe model, we initialized the neural net with an embedding size of 8, a window size of 8, a stride size of 8, and an output size of 128. We trained the model for 30 epochs over the Assemblage training dataset, with the model seeing 10M functions per epoch. Functions were divided into batches of 600; for each batch, 300 unique labels were randomly chosen from the training dataset, and then two functions were randomly selected with each label. We used a learning rate of 0.005, and the Adam optimizer with gradients clipped to [-1, 1]. Per the literature [18; 14], we used $\alpha = 0.2$ as the margin for our triplet loss. REFuSe was trained on three Tesla M40s on an internal cluster, and took 4.5 days to train.

To evaluate the GNN, we used the model checkpoint published by [33] as part of their survey.[1] To evaluate jTrans, we similarly used the fine-tuned model made available on the authors' Github page.[2]

A.2 Evaluation Procedures

We chose to use mean reciprocal rank (MRR) to measure how models performed on our benchmark. MRR, a popular metric in information retrieval, is used to assess systems which take in queries and return a list of possible responses ordered by likelihood of correctness. Letting q be a query, L be the 1-indexed list returned by q, and c be a correctness function, where $c(L[i]) = 1$ if $L[i]$ is a correct response to q and 0 otherwise, q is said to have rank r if the first correct answer in L appears at position r. That is, q has rank r if and only if $1 \leq r \leq \text{len}(L)$, $c(L[r]) = 1$, and $c(L[i]) = 0$ for all $1 \leq i < r$. The reciprocal rank of q is defined to be $\frac{1}{r}$, and the mean reciprocal rank is the average of the reciprocal ranks for every query $q \in Q$. The upper bound on MRR is 1.0 (a correct answer is always in the first position in L), whereas the lower bound on MRR is 0.

In the context of BFSD, q is a query function and L is a list of neighboring functions (embeddings), ordered from nearest to farthest. Due to the large size of our datasets, we used the Hierarchical Navigable Small Worlds [32] approximate nearest neighbor index from Faiss [12] to compute the 30 nearest neighbors to each query function. When no match was found within the first 30 neighbors, we assigned that query an upper bound reciprocal rank of $\frac{1}{31}$ and a lower bound reciprocal rank of 0.

In Section 5.1, we reported the lower and upper bound MRR for the experiments that used our evaluation code. For benchmarking experiments that utilized open-source code from other authors, we reported a single MRR value, keeping with their practice. In particular, when conducting experiments with jTrans, we chose to use the evaluation code published by its authors, as integrating our own code into their codebase was not straightforward. jTrans supports evaluation over multiple pool sizes; in Section 5.1, we report results for pool size $10,000$, as a larger pool size more closely mimics the evaluation methods of the other models. (In our evaluation, the pool is the entire dataset, but we are not limited to having only one function matching the query function in each pool.)

B GNN Common Libraries Details

In our results we stated the significant drop in the GNN's performance on the Common Libraires corpus is due to its inability to handle the variety of functions and function sizes in each application. This is important to verify as the actual cause, as the asperity in the project sizes could easily dominate the results and make it unclear which method actually performs best.

---

[1]This model is available at the following link: https://github.com/Cisco-Talos/binary_function_similarity/tree/main/Models/GGSNN-GMN/NeuralNetwork/model_checkpoint_GGSNN_pair.

[2]This model can be downloaded from https://github.com/vul337/jTrans/.

16
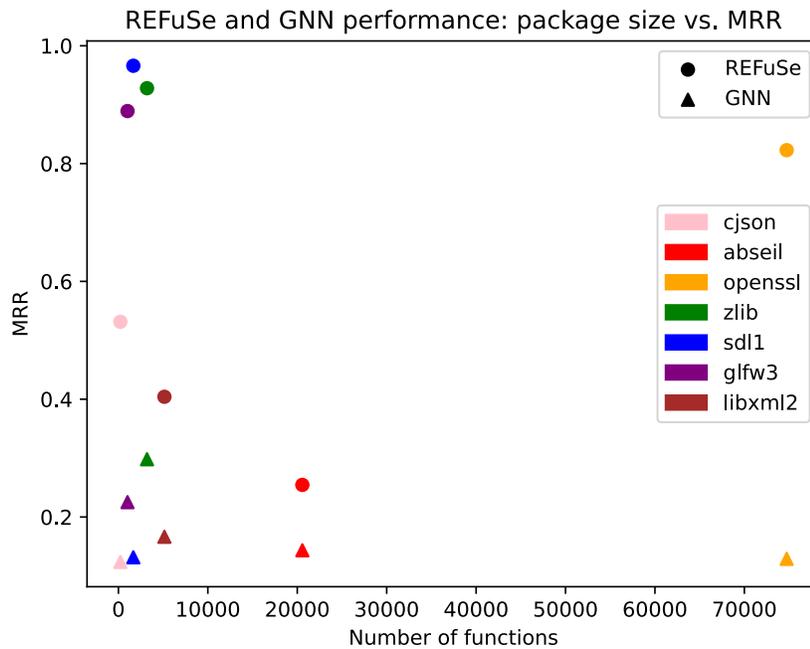
Figure 2: REFuSe and GNN per-package performance. The circles correspond to REFuSe results, while the triangles correspond to the GNN.

We perform this validation in Figure 2, where it can be seen that REFuSe dominates the GNN in performance for each library. Though there are too few libraries to make a definitive conclusion, REFuSe seems to be unfazed by the number of functions in terms of final MRR performance. Yet, the GNN has low performance in all cases and decreases with the number of functions.

17