SUPPLEMENTARY MATERIAL

A DETAILS ON DATASET AND TRAINING LEARNT AFFINITY MODEL

We randomly choose 50K queries from the MSMARCO train data and use TCT»MonooT5 re-ranking pipeline where TCT [9] is a strong dense retriever, and MonoT5 [19] is an effective ranker. We take the top 5 documents from the retriever as positives ($\mathcal{P}_q$), the last 5 as negatives ($\mathcal{N}_q$), and take the top 5 documents from the re-ranked documents using MonoT5 as set S. We release the training dataset with our code.

We conducted our experiments on NVIDIA A100 GPU with 40G of RAM using the PyTorch library. We fine-tuned a bert-base model followed by a linear classifier on our training dataset. We used binary cross-entropy to train our model. We used Adam optimizer with a batch size of 8 and a learning rate 3e-7. We trained our model for 5 epochs. We also used a linear scheduler for the learning rate with warm-up steps. We have released our model's weights along with our code for reproducibility. The affinity model f predicts the similarity between a pair of documents.

B EFFECT OF RELEVANCE AND RETRIEVAL SCORES

According to the definition of the SetAff in Equation 2, the set affinity depends on how the relevance scorer ϕ estimates the query. We study the behavior of adaptive retrieval approaches under different query estimators. Toward this, we use a retriever (BM25) and a ranker (MonoT5) as query estimators. In Figure 6, we report the performances of Gar and Quam with the combination of the estimator as a retriever (denoted with Ret) and a ranker. We observe that both Gar Ret and Quam Ret degrade when the retriever scores are used for the estimated relevance. This shows the retriever scores are not clean estimates of the query. Based on empirical study, we propose to use the ranker as a true estimator of the query.

C EFFECT OF GRAPH DEPTH

We study the performance of different adaptive retrieval approaches when we vary the graph depth k and the batch size b is fixed to 16. Figure 7 and 8 show the performance comparisons for DL19 and 20 respectively.

D EFFECT OF BATCH SIZE

We also study the performance of different adaptive retrieval approaches when the batch size b varies and the graph depth k (=16) is fixed. Figure 9 and 10 show the performance comparisons for DL19 and 20 respectively.

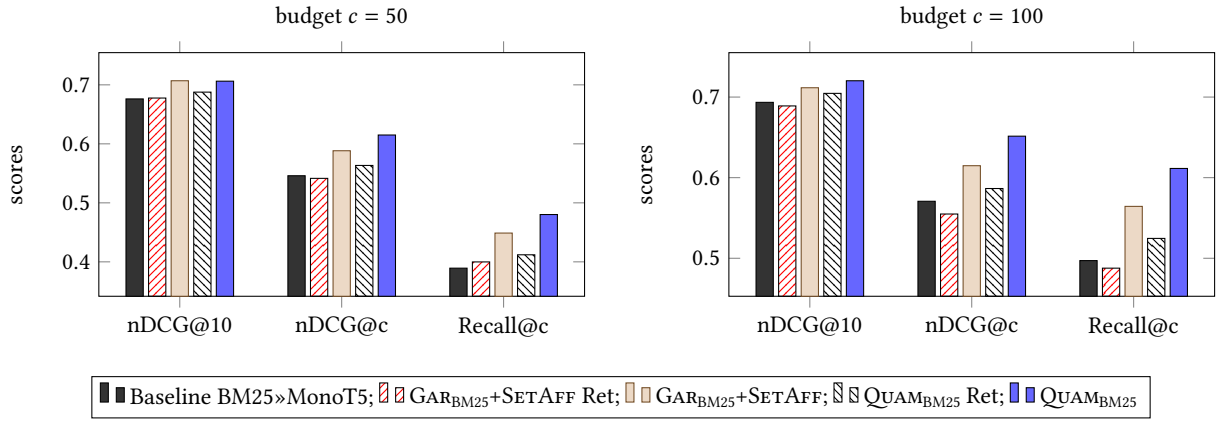

Figure 6: Effect of using retrieval and relevance scores in SETAFF scores on TREC DL19 dataset. Here QUAM$_{BM25}$-Ret and QUAM$_{BM25}$ represent the Query Affinity Model when retrieval and relevance scores are used to calculate the SETAFF scores respectively.

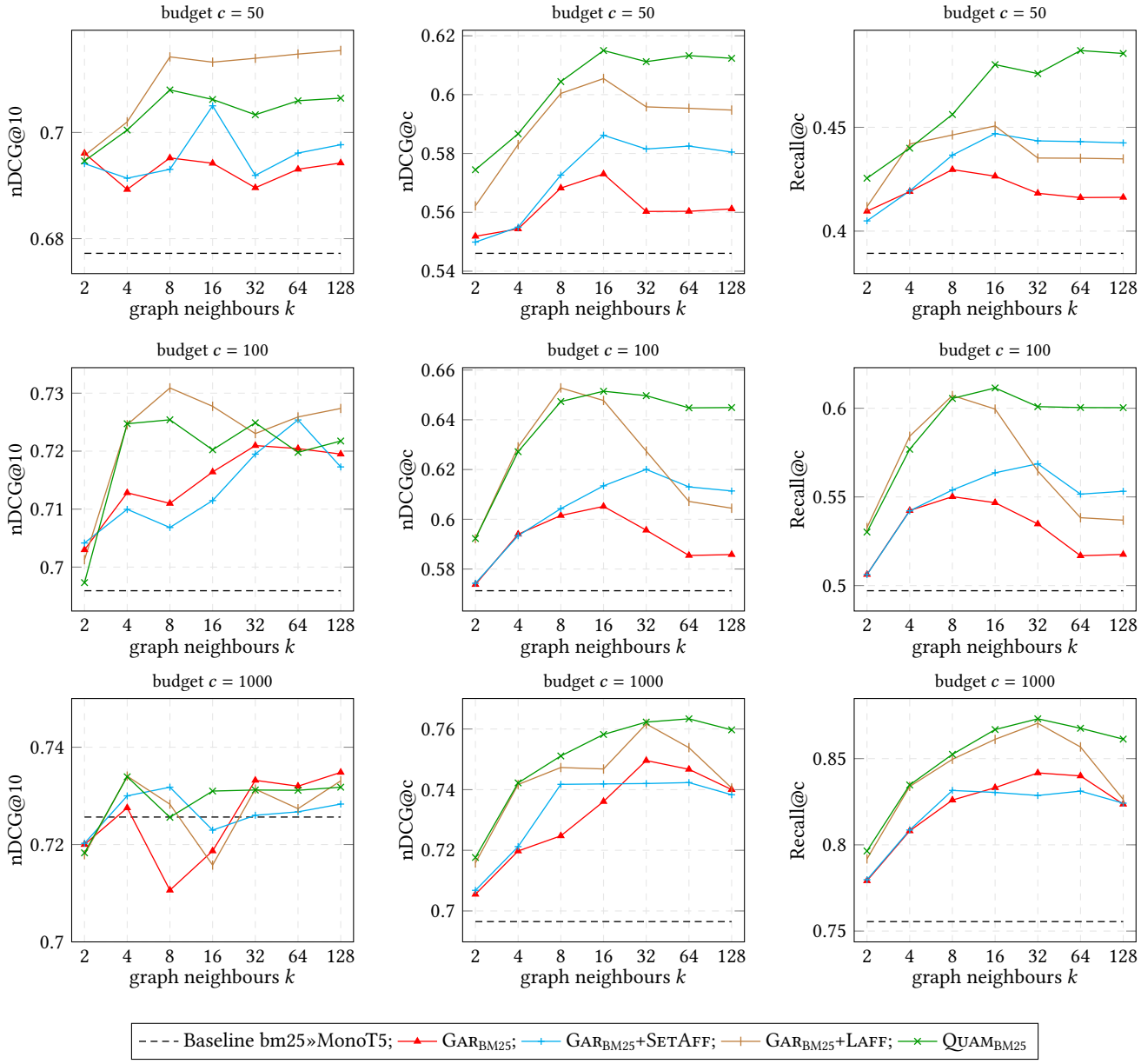

Figure 7: Performance comparison of methods on TREC DL19 dataset when the number of neighbours k vary with batch size $b = 16$.

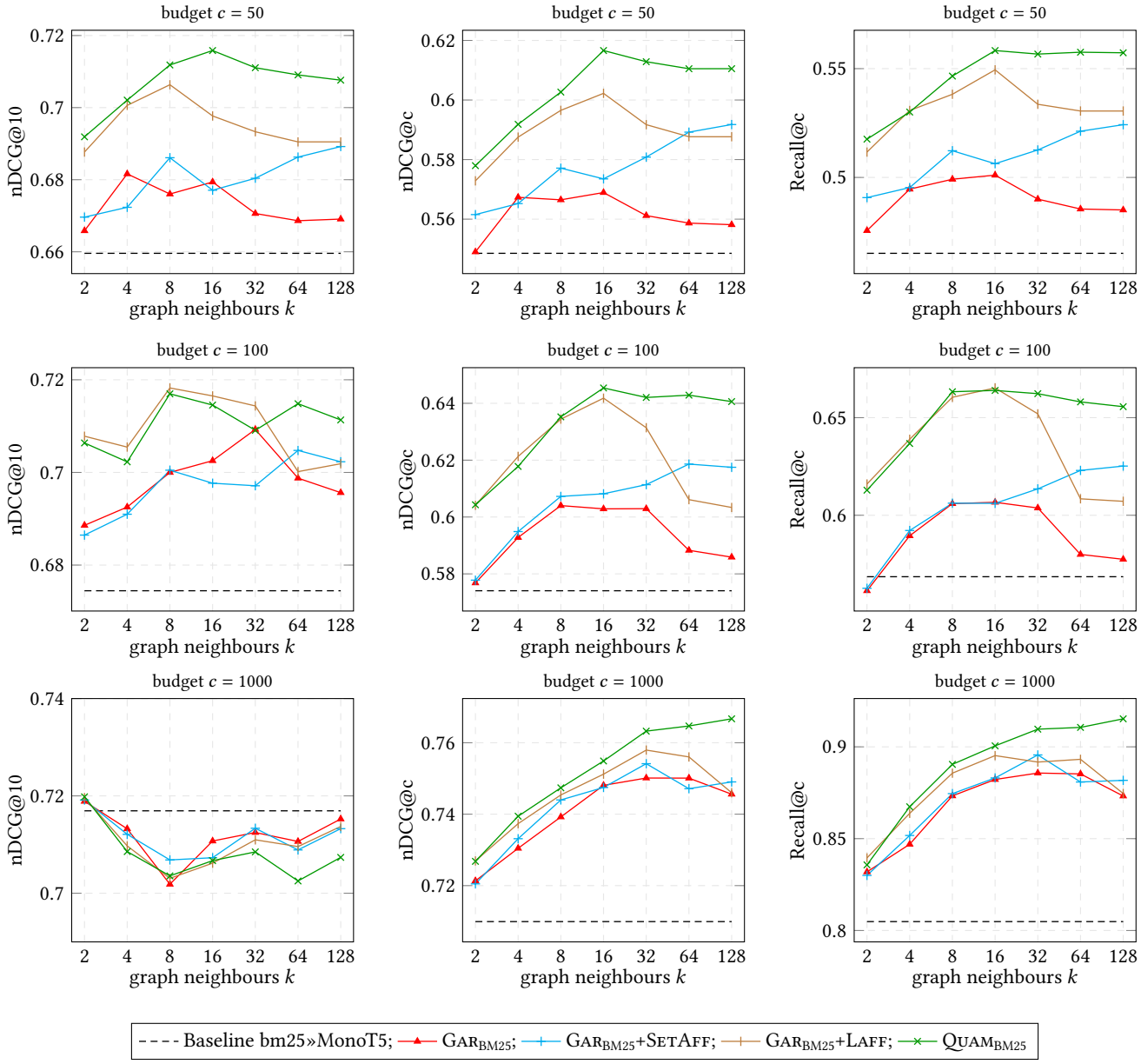

Figure 8: Performance comparison of methods on TREC DL20 dataset when the number of neighbours k vary with the batch size $b = 16$.

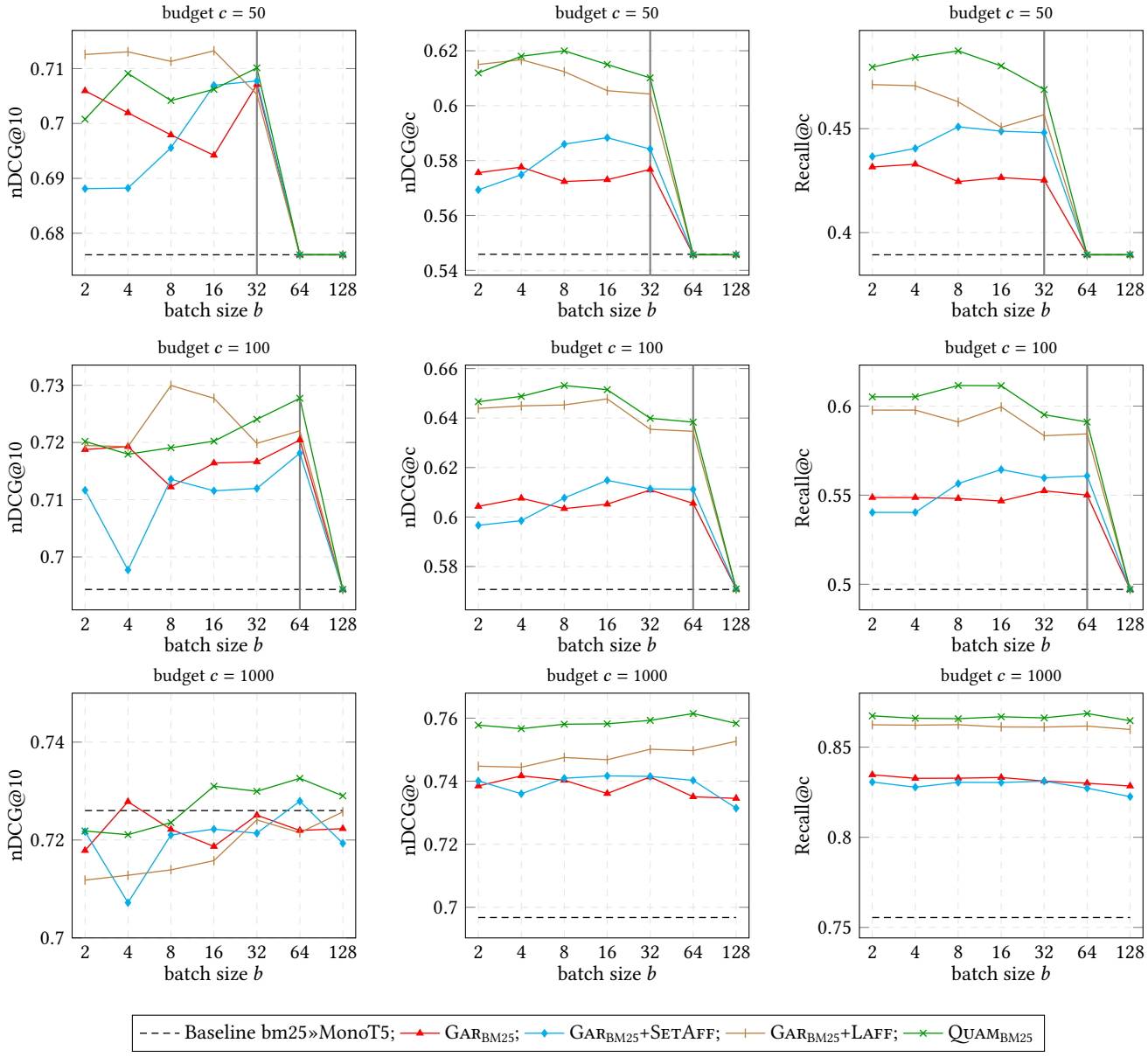

Figure 9: Performance comparison of methods on TREC DL19 dataset when the batch size b vary with neighbors $k = 16$. The vertical lines separate the region where $b > c$.

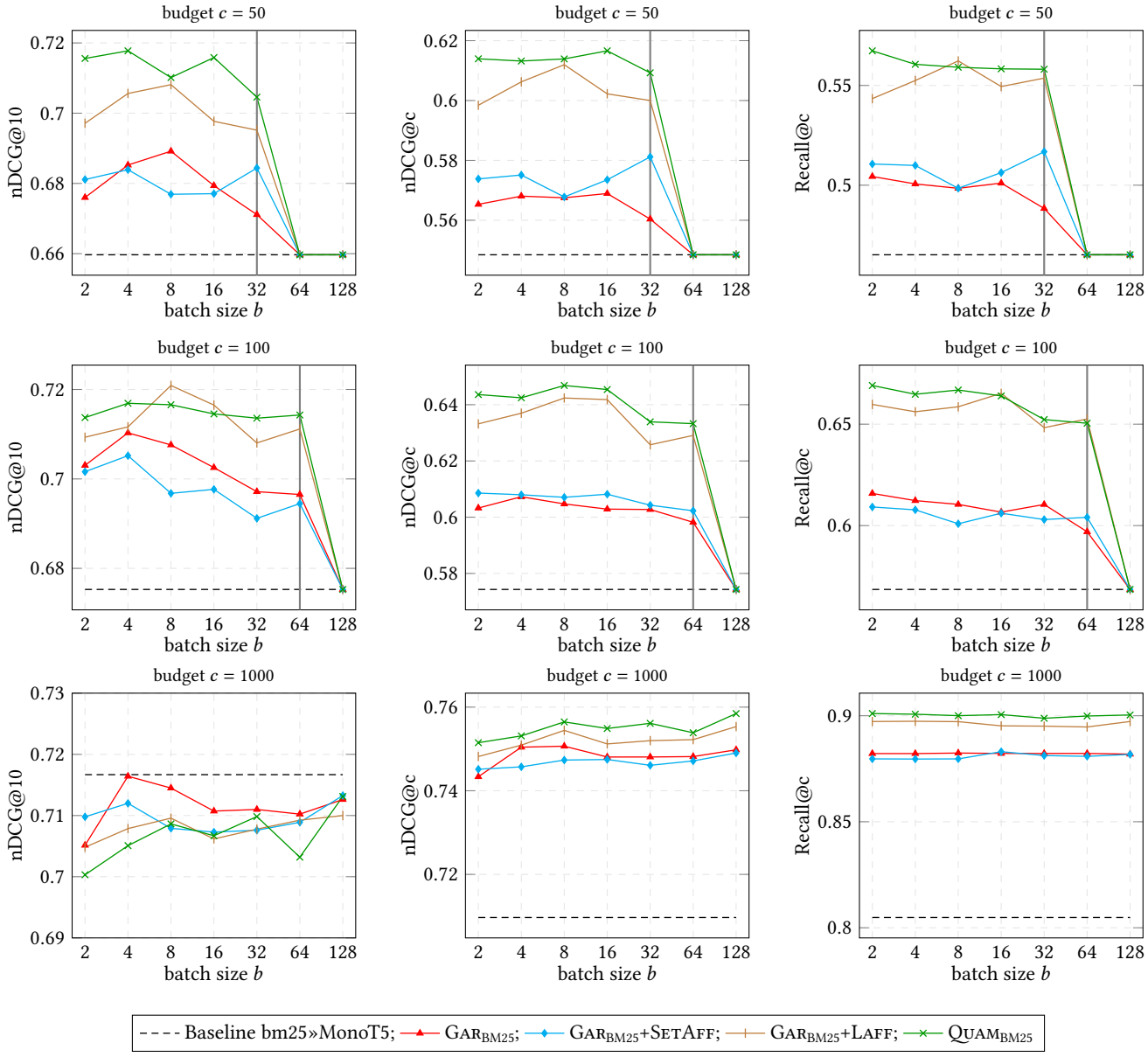

Figure 10: Performance comparison of methods on TREC DL20 dataset when the batch size b vary with neighbors $k = 16$. The vertical lines separate the region where $b > c$.