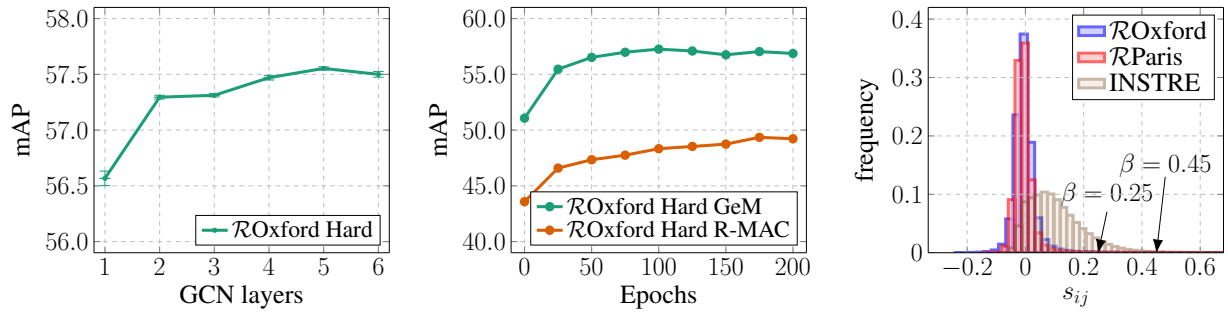

Figure 1: (a) Retrieval performance vs number of GCN layers on $\mathcal{R}$Oxford Hard. (b) Training curves for a two-layer GCN model with GeM and R-MAC descriptors on $\mathcal{R}$Oxford Hard. (c) Binned distribution of pairwise similarity scores s_{ij} for all three datasets.

We would like to thank the reviewers for taking the time to review our work and providing valuable feedback. Here, we address main concerns brought up by each reviewer, and will incorporate minor feedback directly into the draft. We are also in the final stages of refactoring our code repository, and will open source code for all experiments with the final version of this draft.

Reviewer 1 We thank the reviewer for the highly positive feedback and encouraging comments.

Reviewer 2 To address the detailed questions raised by the reviewer, we ran additional experiments to further investigate the properties of our model. Due to time constraints most experiments were done on the $\mathcal{R}$Oxford Hard dataset. Figure 1a shows the effect of adding more layers to the GCN with error bars from ten restarts with random weight re-initialization. We were initially not able to optimize deeper GCNs and thus settled on two layers. However, recently we discovered that adding residual connections (analogous to ResNet) between successive GCN layers significantly improved optimization enabling to train much deeper models. From the figure it is seen that adding layers slightly improves performance from 57.3 with two layers to around 57.6 with five layers. We suspect that further gains can be obtained with more sophisticated optimization techniques and/or architectural modifications analogous to residual connections that aid gradient back-propagation. Figure 1b shows retrieval performance vs training epoch for a two-layer GCN architecture. We see that applying two GCN layers without training (epoch 0) already significantly improves performance of the base GeM descriptors from 38.5 to 51.2. Similar improvements were observed for all other datasets, and we found that normalizing the adjacency matrix according to Equation 2 (in the paper) was instrumental to obtaining this boost. Applying one GCN layer with near identity weights is analogous to "weighted" database side QE, so our results indicate that appropriately normalising the adjacency matrix is highly important for QE and should be further investigated. We also see that training the model with the proposed GSS loss further improves performance by over six points. So both GCN and GSS components are important and best results are generally obtained when the two are combined.

Table 1: mAP on $\mathcal{R}$Oxford 1M.

Method	mAP
GeM	22.7
GeM+αQE	24.2
GeM+DFS	33.2
GeM+FSR	18.8
GeM+DFS+FSR	34.4
GeM+GSS (ours)	**35.8**

Reviewer 3 We have been investigating how to set β automatically, and believe that a promising direction is to use the distribution of the pairwise similarity scores s_{ij}. Figure 1c shows score distributions for $\mathcal{R}$Oxford, $\mathcal{R}$Paris and INSTRE datasets together with β which was set to 0.25 for $\mathcal{R}$Oxford and $\mathcal{R}$Paris and to 0.45 for INSTRE. Here, we see that good values for β tend to be at the *tail* of the score distribution so only the most confident scores get pushed up. This suggests a heuristic to automatically set β by first computing the empirical cumulative distribution function (CDF) of similarity scores, and then setting β to the value where the CDF is sufficiently high such as 0.9. This works well for the three datasets that we evaluated on, and we believe that it can be generalised to other datasets as well.

Reviewers 2 and 3 Both reviewers mentioned varying base descriptors and larger 1M results. Figure 1b shows a training curve for our model with R-MAC [12] image descriptors. R-MAC alone achieves 32.4 on $\mathcal{R}$Oxford which is significantly lower than the 38.5 achieved by GeM. Applying GCN improves the accuracy to 43.6, and GSS optimization produces additional five point gain pushing the accuracy to 49.3 which also outperforms all baselines. These results suggest that our model can be effectively used with different base descriptors regardless of their performance. Table 1 shows results on $\mathcal{R}$Oxford 1M for our model and GEM-based baselines that report results on this dataset. Note that these results are very preliminary as we only had several days to train the model on a much larger dataset. To fit the optimization on the GPU we switched to batch training for the 1M data, where random samples of images were used to compute the GSS loss gradients and update GCN weights. Training to convergence took approximately five hours vs two minutes for the smaller version of $\mathcal{R}$Oxford. From the table we see that our model outperforms the best baseline DFS+FSR by over one point. This indicates that our approach does generalise to the harder setting where the number of distractors is significantly larger. However, as Reviewer 2 pointed out, finding meaningfull clusters is considerably more difficult in this setting so we plan to focus on large scale applications in future work.