We thank the reviewers for their helpful comments. We will move the count-min sketch details to the main paper and add more discussion on experiment results. We will fix the typos pointed out by the reviewers.

To Reviewer 1 and 3:

- *"... EmQL's centroid-sketch representation will not compactly encode any set of sufficiently diverse entities."* and *"assumption that the relevant sets of entities are well described by regions around their centroids in embedding space ..."* In EmQL representation, we assume that elements in the set are similar to each other. While it restricts the power of representing arbitrary sets, this assumption holds in many compositional KB reasoning tasks. With proper pretraining, it enables to accurately retrieve a small subset of candidates and efficiently check their membership with a count-min sketch. Efficiently representing arbitrary sets is a very interesting and challenging task. We would like to leave it for future exploration.

To Reviewer 1:

- *"... if the K contains a row $r_t = [e_r; e_x; e_y]$, the K may be a very huge matric ..."* In this paper, fact embeddings are encoded as a concatenation of its subject, relation, and object embeddings for simplicity. One can easily design more powerful encoding methods to get more compact fact embeddings to save memory. Even though the fact embedding table is large, EmQL is very scalable at inference time with the CPU-based fast Maximum Inner Product Search (MIPS) algorithms, e.g. Faiss (Johnson et al, 2017).
- *"... why the results on MetaQA2 datasets is worse than ... the PullNet method?"* PullNet is a complicated iterative "retrieve and classify" model for multi-hop reasoning trained with distantly supervised labels at intermediate steps. The retrieval on the next step is conditioned on the previous steps. EmQL treats each step independently for simplicity. Intermediate labels are not required to train EmQL.

To Reviewer 2:

- *"Why not compare against Q2Box in KBQA experiments?"* The KBQA problem is modeled as $Y = X.follow(R)$ where R is a weighted set of relations. Query2Box embedding is designed to follow a specified relation r. It's not clear how to follow a set of relations R with learned relation weights.
- *"... some experiment to judge the efficacy of count-min sketches to represent sets ..."* The entailment experiment results in Table 2 show the ablation study without sketch (EmQL - sketch). Without the count-min sketch, the model performs more than 30% worse in Hits@3 on complicated queries, and 10% worse on simple queries. We run the set decoding experiments proposed by the reviewer. On FB15k-237, we get 99.8% F1 on sets with less than 100 elements (with k = 1000).
- *"When vacuous sketches are used in the intermediate steps ... what is the intermediate output?"* Correct, the intermediate output is the dense-sparse representation of entities returned from the top k facts. When k is large, we should increase the width of the count-min sketch to accommodate more entities. In general, it should be larger than 2 * k to minimize the risk of collision. Increasing k helps improve the recall but can decrease the efficiency.
- *"Both union and intersection of two sets have the same centroid and with a vacuous sketch . . . "* Vacuous sketches are often applied on relations in KBQA tasks when relations on the inferential chain are learned. For union (intersection) of two sets of entities, the sketches of the two sets are usually present from the previous steps. The sketch of the unioned (intersected) set is then derived from the two sketches. It is possible to get a vacuous sketch for the unioned (intersected) set in case that the sketches of both sets are vacuous. We would expect performance to drop in such cases.
- *"... however EmQL first extracts top k elements (k is fixed), irrespective of the size of the set ..."* and *"This becomes an issue when the ground truth is a set of different cardinality for different queries . . . "* An alternative of top k retrieval is to apply soft thresholding to the ranking scores of all elements. This requires computing ranking scores for all entries in the KB and can be very inefficient when KB is large. We compromise with the top k operation that is fairly efficient with the CPU-based approximate Maximum Inner Product Search (MIPS) (Johnson et al, 2017).
- *"EmQL can be used for the task of KBC . . . "* Yes, it's interesting to experiment with EmQL on KBC tasks. We would like to leave this for future work.

[1] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. arXiv preprint arXiv:1702.08734, 2017