We thank the reviewers for their time and valuable feedback. Overall, we are glad that the reviewers found BetaE to be a novel contribution to advance reasoning on KG. Below, we clarify a number of important points raised by the reviewers.

Multi-modality (R3, R4). Reviewers raise concern on multi-modal embeddings. We agree that the Beta embeddings along each dimension can be at most bi-modal. We alleviate this problem by introducing multiple independent Beta distributions and learn high-dimensional embeddings. We will highlight this limitation in Sec. 4.3 and plan to further improve the handling of multi-modality and use mixture models in the future work.

Comparison with additional baselines (R3). R3 suggests that "the authors can adapt the FOL queries to other multi-hop reasoning models". We argue the differences in tasks and setups below. The goal of [1] is to use first-order logic (FOL) rules to improve knowledge graph completion (given an entity and a relation, find the target), while our setting is different as we aim to *find all entities that satisfy a given FOL query*. Another key difference is that [1] needs to model all the intermediate entities in a FOL rule, which is expensive in multi-hop query answering (exponential complexity). For [2], they use an LSTM to model path queries. One limitation is that an LSTM can only be applied to answering path queries (as R3 mentioned) and it is not obvious how to extend [2] to handle more complex query structures with intersection, union, and negation. Another difference is that our main contribution lies in the direction of a **novel probabilistic embedding space** with the ability to handle any FOL operations and [2]'s contribution is a **more powerful neural network architecture**. To be concrete, we further combined our model with the LSTM architecture and trained and evaluated the model only on path queries (1p/2p/3p) on NELL. We are able to further improve BetaE's performance using the LSTM architecture (25.5 v.s. 23.8 average MRR). In conclusion, the contribution of [2] and ours are orthogonal and we plan to combine BetaE with more powerful architectures, e.g., GNNs or Tree-LSTMs, in our future work. We will clarify this point in the final version of the paper.

Queries without answers and non-ranking tasks (R3, R4). R4 raises concern on whether BetaE can further handle queries without answers. In fact, it is an open problem for all the embedding-based methods (including those designed for link prediction) to model empty sets or to answer whether an existential query is true or false. Here we investigate one potential solution. Since BetaE can effectively model the **uncertainty** of a given query, we can use the differential entropy of the query embedding as a measure to represent whether the query is an empty set (has no answers). For evaluation, we randomly generated 4k queries without answers and 4k queries with more than 5 answers for each of the 12 query structures on NELL. Then we calculate the differential entropy of the embeddings of each query with a trained BetaE and use this to classify whether a query has answers. As a result, we find an ROC-AUC score of 0.844 and list the ROC-AUC score of each query structure in the Table below. These results suggest that one promising idea is to use the differential entropy of the learned embedding as an indicator to whether a query contains answers. Note that BetaE naturally preserves this property since (1) we did not explicitly train BetaE to optimize for correlation between the differential entropy and the cardinality of the answer set; (2) we did not train BetaE on queries with empty answers. It means BetaE's performance can be further improved with additional supervision. We will further add these points to the paper and we believe this is also an additional justification to support our uncertainty claim, a concern pointed out by R3. We also thank R3 for the insightful suggestion.

1p	2p	3p	2i	3i	pi	ip	2in	3in	inp	pin	pni
0.825	0.766	0.793	0.909	0.933	0.868	0.798	0.865	0.93	0.801	0.809	0.848

Theoretical analysis on BetaE expressiveness (R3). R3 brings up an important question whether BetaE is fully expressive and can model any given query-answer pairs on a KG. This is a challenging problem and our ongoing research concern. Our aim is to construct entity embeddings as well as probabilistic logic operators, such that for any valid FOL query q, we have $\text{Dist}(v;q) < \tau$ if $v \in [q]_Q$ and $\text{Dist}(v;q) \geq \tau$ if $v \notin [q]_Q$, where τ is a threshold.

1p queries/link prediction (R4). R4 asks for experiments on 1p queries (link prediction). We ran the experiments on NELL and the MRR results of BetaE (trained on complex queries), BetaE (trained only on 1p queries) and TransE are 0.404, 0.383, 0.376. Note that we follow the convention: averaging over query-answer pairs as in the link prediction literature. We do not expect a large gain since our target task is modeling more complex multi-hop logic queries and supporting all FOL operations. Moreover, we find BetaE benefits from additional training on complex queries.

Other comments. We will put the MRR tables for all the datasets in the main body and defer the H@K tables to the Appendix for consistency of the evaluation metrics. Note that MRR shows a similar trends as H@K (R1). We used the filtered setting for all experiments (R3). We will make explicit the input/output of the model in Sec. 1 as well as add more discussions in Sec. 5.3 on differential entropy and query uncertainty including its definition and the evaluation metrics Spearman's coefficient (R3). In order to model containment, one direction is to have an explicit likelihood threshold and we view regions that have density greater than the threshold as the effective support of the query or the entity. Then we can tune the threshold and check whether the support of the answer entities are enclosed in the query (R4). We will further discuss the difference between our logical operators and their real counterparts in Sec. 4 (R4).

- [1] Knowledge graph embedding with iterative guidance from soft rules
- [2] Chains of reasoning over entities, relations, and text using recurrent neural networks