We thank reviewers for providing such insightful feedback and detailed assessments of our submission. We will address all concerns and other technical questions below, and revise and update the submission. The code link issue is fixed, please check here for reference.

Q1: [R1, R2, R3] How to encode texts and what is contextual information? **Ans.** The texts are from label names, synonyms and descriptions or annotations of the concepts. Specifically, we use the texts(tokens) from these meta-information for all training objectives.

Q2: [R1] Technical soundness questions **Ans.** (1) We use two transformers to identify different contextual structural information. Having two transformers, even if they share parameters, allows these attention mechanisms to focus on different aspects (local and global) of the ontologies. (2) Yes, triplets are given in the ontologies. (3) subclass set only contains sub-classes of the concepts. (4) Following BERTMap [1], concepts which share a common super-class are disjoint, which is used to generate hard non-synonym pairs. (5) The relation normally contains one or two word tokens, where the information density is smaller than entities. Thus, we use the pooling function to compress the relation features to a global representation. (6) Eq 8 is to predict relation for each triplets. (7) Eq 9: same set of concepts may generate different paths, so we sum over all concepts and paths. (8) In path-level encoding, we consider to only learn concepts representations in paths for fast training. (9) We predict less or equal candidate concepts for fast inference. (10) Structural relations in Bio-ML data are limited, so we mainly use subclass, disjointWith and synonyms in our settings. (11) The examples of tags are 'rdf:ID' and 'rdf:resourc', but their label names may be the same.

Q3: [R2] Readability and Technical soundness questions **Ans.** (Readability 2) $f(\cdot)$ is the feature extraction module and the outputs are embeddings. (Readability 3) In our training settings, we consider [mask] and [sep] are the same. (Technical soundness 2) We train LaKERMap on all ontolgoies in a self-supervised manner. Given a concept from any ontology, we generate the embeddings and predict equivalent concepts from the candidate sets.

Q4: [R2, R3] Triplet contrastive learning related questions **Ans.** This contrastive loss function encourages the model to produce similar embeddings for positive pairs (like (h,r,t) and (h,r,[mask]) when mask is t) and different embeddings for negative pairs (like (h,r,t) and (h,r,[mask]) when mask is not t). This helps the model better differentiate between correct and incorrect entity-relation pairs. Moreover, it allows the model to understand the context from both directions, thus capturing a more comprehensive understanding of the relationships and properties of concepts.

Q5: [R4] Weaknesses and questions **Ans.** (1) In this work, we mainly consider the ontology matching problem for equivalence matching and will extend to subsumption matching in the future. The main challenge of ontology matching systems are fast mapping generation for large ontologies. Our proposed LaKERMap has significant improvements in term of inference time. (2) We use the pre-train language model parameters as our initial parameters, and fine-tune it with our proposed multiple training objectives. (3) We encode hierarchy structures information of a ontologies as path-level representations, and we self-supervised train on all ontologies. In this way, we do not need reference alignment to generate path between ontologies.

References

[1] Y. He, J. Chen, D. Antonyrajah, and I. Horrocks. Bertmap: A bert-based ontology alignment system. *arXiv preprint arXiv:2112.02682*, 2021.