Response Letter

Dear Editors,

We sincerely appreciate the time and effort of the reviewers in evaluating our manuscript, *"Towards Interpretable Embeddings: Aligning Representations with Semantic Features."* Their thoughtful comments have significantly contributed to improving the clarity and precision of our work. Below, we provide a detailed response to Reviewers' comments, outlining the corresponding revisions that are included in the revised paper.

Response to Reviewer 1

We sincerely thank the reviewer for their constructive feedback, we provide a point-by-point response to address each comment below.

Interpretability Definition: We greatly appreciate this insightful suggestion regarding the need for a definition of interpretability in the context of our work. As suggested, we have added a clear definition of interpretability in the Introduction section (lines 35-38) to clarify the concept as it pertains to our method. This definition establishes how InterpretE approaches interpretability, particularly in the context of making knowledge graph embeddings more interpretable through the alignment of embedding dimensions with human-understandable semantic features.

Ontology vs. Knowledge Graph Embeddings

We fully agree with the reviewer's observation regarding the relationship between knowledge graphs and ontologies. A knowledge graph can indeed be viewed as the ABox of an ontology, and therefore, ontology embedding methods are also applicable to knowledge graph data. Consequently, we have removed the statement from Section 3.3 that incorrectly suggested ontology embedding methods do not address knowledge graph data. Instead, we have clarified the distinction between our approach and existing ontology embedding methods in the following paragraph to accurately reflect our contributions.

Section 5.1 Notation and Definitions

We appreciate the reviewer's attention to the consistency of definitions and notation. In response to the issues raised:

- **Knowledge Graph Definition:** We have updated the definition of "knowledge graph" in Section 5.1 to align with the definition in Section 2.1, ensuring consistency throughout the manuscript.

- **Explanation of R_C:** We have now provided a clearer explanation of R_C and its distinction from R, which resolves the confusion regarding this notation.

- **Difference between P(r | class(head) = C) and P(r,v | class(head)=C):** We have added clarification regarding this difference and explicitly defined "head" to remove any ambiguity.

Feature Vector f_e Clarification

We appreciate the reviewer's query about the feature vector f_e and acknowledge that the equation was not represented in the best way. We have now modified the equation and made it more generalizable and clearer. Overall, we have refined Section 5.1 to add better explanations and make it easier to comprehend.

Notational Issues with u'_e

We acknowledge the reviewer's observation about the inconsistent use of the symbol ′u'_e. The symbol u'_e(r,v) denotes the value for the new InterpretE embedding, which is associated with a relation and a value. On the other hand, u'_e refers to the final InterpretE embedding associated with the entity e. We have revised the notation to ensure clarity and to distinguish between these two uses.

Impact of InterpretE on Downstream Applications

Thank you for this thoughtful question, which also aligns with concerns raised by another reviewer. We would like to clarify that InterpretE was designed to map knowledge graph embeddings (KGEMs) to human-interpretable semantic features, and thus, complementing rather than replacing existing KGEMs. Its focus is on tasks where semantic understanding is essential, such as those in Figure 16, which directly relate to semantic similarity. These applications benefit from embeddings that reflect human-understandable features, a capability that standard KGEMs do not inherently provide.

Since tasks like link prediction and query answering have been extensively optimized for KGEMs, we have not evaluated InterpretE on them, as they fall outside its intended scope. Instead, InterpretE offers a way to extract structured semantic information from existing models without the need for retraining embeddings from scratch. While KGEMs remain the most suitable choice for tasks like link prediction, InterpretE is particularly useful in scenarios where interpretability is a key requirement. We acknowledge that the broader impact of InterpretE embeddings on downstream tasks beyond interpretability-focused applications remains an open question, and we would like to investigate this in future research.

To clarify the above distinction, we have updated the manuscript and acknowledged that existing KGEMs remain the best option for such tasks. We have also included in multiple sections that feature selection in InterpretE is motivated by the specific semantic tasks it is intended for. This

addition highlights that the selection of task-relevant features is essential for achieving interpretability in the context of semantic tasks.

- We have corrected the typo in Equation 1 and we have corrected the term "OWL2VecVec*" to "OWL2Vec*" as per the reviewer's suggestion.
- **References to ArXiv:** We have now updated the references to include the published versions of the works cited where available.

Response to Reviewer 2

We sincerely appreciate the detailed and thoughtful feedback provided by the reviewer. Your insights have been invaluable in improving the clarity and rigor of our manuscript. Below, we provide a point-by-point response to address each comment. We hope the revisions address all concerns satisfactorily, and we truly appreciate your valuable input in improving this work

Figures Readability : We acknowledge that some figures were too small for readability in print format. We have improved the figures by increasing the text size to ensure clarity in both digital and printed formats.

Polysemanticity and Feature-Based Approach

We deeply appreciate the reviewer's insightful and thoughtful comments on the paper. We completely agree with the reviewer that, similar to word2vec embeddings, KG embeddings are inherently polysemantic, as highlighted in the literature ([14]). This important point is indeed mentioned in page 2, lines 29-32 in the introduction, where we motivate the work. We would like to emphasize that the intention of our work is not to consider polysemanticity as a disadvantage of the embeddings. On the contrary, as the reviewer astutely pointed out, this polysemanticity enables generalization across various tasks, which is a significant advantage.

At the same time, we aim to bring awareness to the fact that this very feature—polysemanticity—makes the embeddings less suitable for certain semantic tasks, particularly when it comes to semantic similarity. Our work highlights the fact that, while KGEMs are excellent for many tasks, they are often not ideally suited for tasks that rely on semantic representativeness in embedding vectors. It is this gap that we hope to bring to the attention of the research community.

In this context, InterpretE is not designed to replace existing KGEMs. Rather, we intend it as a tool to derive interpretable embeddings from these models, specifically for applications where the embeddings are used for semantic tasks. We envision scenarios where human-understandable features are essential, and InterpretE aims to provide those interpretations while leveraging the existing models. Since InterpretE is designed for interpretability rather than predictive performance, we have not conducted link prediction evaluations, as this is outside the scope of its intended use.

InterpretE is proposed as a method to derive monosemantic representations from existing KGEMs, with the flexibility to use as many or as few dimensions as the desired features, as dictated by the intended semantic task, without the need for the costly process of training new embeddings from scratch. To reflect this, we have updated the introduction in lines page 2, 39-41 and page 3, 10-15

We hope this response sufficiently addresses the reviewer's concerns and clarifies our intended approach. Thank you once again for your valuable input, which has contributed to enhancing the clarity of the paper.

Handling of One-to-Many Relations

We appreciate the reviewer's attention to this detail. We would like to clarify that one-to-many relations were never excluded, the original statement merely referred to the fact that unique values define feature categories—meaning that repeated values across multiple triples collapse into a single category.

For example, for all organization entities and relation *isLocatedIn*, all triples with value as Paris, (regardless of the head entity) only one `category' for Paris is created, rather than multiple redundant ones. To avoid confusion, we have rephrased this statement for clarity.

Feature Selection and LLM-Based Automation

We would like to sincerely thank the reviewer for their insights into the potential of LLMs for automating the feature selection process, and we acknowledge the promise of LLM-based approaches for feature extraction and that refining prompts and leveraging few-shot learning or fine-tuning could improve results. At the time of conducting this work, however, LLM prompting techniques were not yet mature enough to provide the transparency and statistical grounding required for interpretability in the context of KGEMs. Furthermore, LLMs inherently introduce opacity and complexity, which run counter to our primary goal of ensuring that the features in the embeddings are human-understandable and interpretable.

Our work focused on deriving monosemantic, interpretable features from the KG dataset, and this is directly in contrast to the polysemantic nature of features generated by LLMs. We realized that relying on LLMs would introduce significant challenges in understanding the reasoning behind the extracted features, thereby compromising the clarity and interpretability that is central to our approach. Therefore, while we did perform limited exploration LLMs, we quickly determined that pursuing this approach would not align with the objectives of our work. As shown in Figure 15, the answers generated by the LLMs were not statistically driven, further reinforcing our decision. Although LLMs have great potential in other contexts, for this specific task, we felt it was more critical to focus on methods that prioritize interpretability and transparency.

To address the reviewer's concerns and keep the focus of the work tight, as suggested by another reviewer, we have condensed the discussion of LLMs and included this in Section 4.3 (for feature extraction) and Section 6.4 (for similarity evaluation) We have moved the full details of our experiments to supplementary material, available on the GitHub repository

(https://github.com/toniodo/InterpretE) for those who are interested in the techniques we explored. We have also clarified that the input to LLMs is still a knowledge graph, but it is represented in a document format, similar to how KGEMs take the KG as input in the form of text files with triples. We have updated the text to make this distinction clearer.

We are very grateful for the reviewer's suggestion and plan to revisit this research direction in future work as LLM-based methods continue to mature. This is discussed further in Section 7 of the paper. We hope this response clarifies our approach and sufficiently addresses the reviewer's concerns. We look forward to further exploring these ideas in future research.

Reflections on [14] and their approach

We thank the reviewer for highlighting the relevance of [14] and for the insightful suggestion. We indeed found Cunningham et al.'s work fascinating, particularly in how it derives monosemantic features through a sparse autoencoder. However, there are key differences in our motivation and approach for deriving interpretable features from KGEMs. While Cunningham et al. reverse-engineer features from an existing model, our goal with InterpretE is to align the embedding vectors of knowledge graph entities with a set of user-defined or task-specific aspects. Unlike their approach, which doesn't offer customization, InterpretE focuses on aligning the features to user-dictated or task-driven features that can be tailored to specific needs. Our approach focuses on generating new, interpretable embeddings that reflect these desired aspects, rather than just extracting features from pre-existing embeddings. The experiments in Section 6 demonstrate how InterpretE can derive embeddings aligned with various combinations of entity aspects. We have updated Subsection 3.1 to further clarify this distinction.

Nevertheless, we agree that Cunningham et al.'s approach offers exciting possibilities, and we are currently exploring this line of research to see if it can be applied to KGEMs as well. We hope to achieve promising results in this direction in future work.

Other Corrections

- **p.2, l.32:** We have clarified that the lack of direct correspondence between entity aspects and embedding dimensions is not necessarily a flaw but rather an inherent characteristic of distributed representations. This point is now explicitly discussed.

- **p.3, l.46:** The definition has been updated to consistently use the same notation for the KG throughout the paper

- **p.6, l.12 & l.20:** OWL2Vec is now mentioned earlier, and OWL2VecVec has been correctly formatted.

- **p.8, l.43:** We appreciate the reviewer's insightful suggestion. Indeed, introducing a hierarchy of relations, such as making *playsFor* a subrelation of *isAffiliatedTo*, would have allowed us to include both relations in our experiments. However, even with this solution, the entities and values associated with the two relations would have largely overlapped.

Since our goal is to extract unique and interesting information from the knowledge graph datasets, we opted not to consider both relations in order to maintain diversity in our experimental settings. That said, we agree that including both relations would not have posed any significant issue for the method, and we appreciate the potential value of such an approach for future work.

- **p.8, l.46:** Thank you for pointing this out. The phrase "These values, coupled with the associated relation, serve as the entity aspects for the experiments" was intended to convey that the most represented values for a given relation, along with the relation itself, were selected to serve as entity aspects or features in the InterpretE experiments, as described in Section 5. We have revised this statement to provide greater clarity and ensure the intended meaning is clearer (page 9, line 39)
- p. 11, l 27 - The phrase "is computed based on its occurrences in triples" refers to how the frequency of an entity or value is determined by counting how often it appears in the triples of the knowledge graph (KG). Specifically, for a given relation in the KG, the number of times an entity or value appears as part of a subject-predicate-object triple is used to calculate its occurrence count. This frequency count helps in selecting the most represented values for the experiments, which are then used as entity aspects in InterpretE. We have rephrased this section to ensure this process is described with greater clarity.
- **p.11, l.18, 26, 30:** We have introduced V_r properly before its first use, clarified that v is not actually used in the formula, and explicitly defined R_C and 'head'.

- **p.11, l.46 (definition of f_e):** Thank you for pointing this out. The notation has been generalized to accommodate an arbitrary number of values and make the equation more clear and understandable. Overall, we have refined Section 5.1 to add better explanations and make it easier to comprehend.

- **p.12, l.6:** D now explicitly depends on C, and the notation has been updated to D_C in both text and algorithm for clarity.

- **p.12, l.32:** The optimization function has been corrected to properly bind both w and bias term b, ensuring mathematical consistency.

- **p.13, Algorithm 1 (lines 7 & 17):** The operation previously denoted as 'union' has been replaced with an explicit CONCAT function to avoid ambiguity regarding vector concatenation.

- **p.13, l.29:** Fixed "entities" → "entity."

- **p.14, l.13:** Closing bracket added.

- **p.15, l.42:** Adjusted wording to correctly reflect that the kernel trick generally maps data into a much higher-dimensional space, not just an additional dimension.

- **p.16, l.43:** We appreciate the reviewer's keen attention to wording. The statement about LLMs having "finite knowledge" has been revised for precision. We now clarify that while LLMs contain vast but bounded pretraining knowledge, their primary limitation is the lack of structured, dynamically updated world knowledge specific to KG datasets.

Response to Reviewer 3

We sincerely thank the reviewer for their constructive feedback. Below, we address each point raised in the review.

Mathematical Notation in Section 5.1: We acknowledge the reviewer's difficulty in following the mathematical notation in Section 5.1. To improve clarity, we have revised and simplified some of the equations and provided additional explanations for key concepts. We also appreciate the reviewer's suggestion for more clarity - A short introductory paragraph has been added before the formal explanation. This paragraph introduces the key concepts and provides an intuitive explanation of the feature selection process, with an example to guide the reader. Overall, we have refined Section 5.1 to add better explanations. We hope these changes make the section more accessible and easier to follow.

Impact of \tau on Downstream Results:

We acknowledge the reviewer's concern regarding the impact of the threshold \tau on downstream results. In our analysis, we focused on the most frequently occurring relations and did not explicitly measure the effect of varying \tau. However, we did observe that even underrepresented relations, such as "gender" in YAGO3-10 (~7% of persons), still yielded high scores. The rationale behind setting the threshold was to minimize uncertainty in the hyperplane when training the SVM, ensuring that the larger amount of data contributed to a more generalizable model.

To address this concern more rigorously, we recognize that a more systematic evaluation could involve testing extremely underrepresented relations (e.g., those associated with fewer than 100 entities) and measuring the kappa score on the test set. This would provide a clearer understanding of how varying \tau affects generalization and model performance. We would take this into consideration for future work.

Relations Left Out for Given Entities:

In response to the reviewer's question, we avoided using relations with fewer than 100 entities linked to them. The number of relations excluded varied depending on the dataset and entity

class. Typically, we retained relations that accounted for 70-80% of the triples, with this proportion varying based on the class and dataset in question.

Clarification of Example in Section 5.1 (Page 11, Lines 37-38):

We have reworked this section to improve clarity. The example of abstraction has been discussed in detail at the end of section 4.2, therefore, it has been removed from here to avoid confusion.

Algorithm 1 (Line 2: U' not in math mode): This has been corrected to maintain consistency with the rest of the manuscript.

Section 7: We acknowledge the reviewer's concern regarding Section 7's alignment with the rest of the paper. To address this, we have relocated the detailed content of Section 7 to a supplementary file, now accessible in our GitHub repository at https://github.com/toniodo/InterpretE . This ensures that the main manuscript remains focused while providing interested readers with comprehensive information. At the same time, we have incorporated a brief discussion in subsections 4.3 and 6.4 to summarize the motivation and limitations of that work, thereby maintaining coherence in the core content of the paper. We hope this response sufficiently addresses the reviewer's concerns.