We thank the reviewers for their comments and feedback. We believe issues of language, terminology, and organization can be fully addressed in the revision. We address specific issues below.

1 Reviewer 2

Why is the slowness of current graph isomorphism algorithms relevant to the problem of producing isomorphism-injective graph representations?

Producing isomorphism-injective representations for graphs solves the graph canonization problem, which in turn solves the graph isomorphism problem. Thus, the runtime of the fastest graph isomorphism algorithm is a upper bound for the runtime of the fastest isomorphism-injective function. Thus, we focus on *multivalued functions* rather than on *functions*.

Page 3: the comment that the running time in algorithm is a function of the input seems odd. The run time of any algorithm will be in terms of the input

We will rephrase this comment to make it clearer. It is not the run time per se that we want to comment on, but rather that the theorem is contingent on the *number of recursive applications* of the RNN. However, it is hard to allow the RNN to recursively run on input for a variable number of steps. In contrast, the theorems cited for NNs as universal function approximators are true for only one application of the NN.

Page 4: The justification for Postulate 1 is not very clear

We will clarify the postulate and the surrounding discussion. We mean non-induced subgraphs and the O should be a Ω.

Algorithm 1: what is d_c? And what does an extended function mean?

d_c is just the dimension of the encoding. An extended function $g : \mathcal{A} \to \mathcal{B}$ of $f : A \to B$ is such that $A \subset \mathcal{A}$ with $g|_A = f$. We will clarify this in the text.

2 Reviewer 3

Algorithm 2 needs to be analyzed in detail for runtime. In particular, Corr. 1 suggests that Alg. 2 has superpolynomial runtime, that there is a second paper hidden here (in case you are faster than Babai), or that the claim of Corr. 1 is incorrect

Corr. 1 is in terms of $\mathbf{C}$ which is the multivalued function variant of C as per Definition 7, therefore it does not suggest super-polynomial runtime. We will clarify this in the text.

Regarding Definition 1 and Lemma 1

We will clarify this in the text but with graphs of bounded size we mean bounded in terms of number of nodes, number of edges, and that the label functions are bounded.

I am not sure how Remark 1 results from the argumentation below Thm. 5. Furthermore, it seems unclear to me why the point p to which the identified subsequence converges (called $p = Alg([G]^*)$) would actually be in $img(Alg)$, i.e., does there exist a graph H such that $p = Alg([H])$?

That's correct, we do not assume that $p = Alg([G]^*)$ is in $img(Alg)$, we will change the notation to make this clearer. We are simply deriving that convergent subsequences cannot be avoided, since $\boldsymbol{\mathcal{G}}$ is infinite and the image is bounded.

3 Reviewer 4

Proofs of Theorems 1, 3, and 6 follow from basic results in undergraduate topology classes

We agree and we do not consider Theorems 1, 3, and 6 as the contributions of this paper, but rather Theorems 7, 8, 9 and 10. We invite future work to investigate other convergence than pointwise convergence.