We thank all the reviewers for their comments about the novelty and significance of the work. Reviewers all had constructive suggestions that will improve this paper. Below we address reviewers' two common comments.

Comparing with other baselines and why we mainly emphasize on comparing to NeuralLP: To the best of our knowledge, NeuralLP and our method are the only scalable[1] and differentiable methods that provide reasoning on KBs without needing to use embeddings of the entities at test time, and provide prediction solely based on the logical rules. Other methods like NTPs [1] and MINERVA [4], rely on *some type of learned* embeddings at training and test time. Since rules are interpretable and embeddings are not, this puts our method and NeuralLP in fully-interpretable category while others do not have this advantage (therefore its not fair to directly compare them with each other). Moreover, methods that rely on embeddings (fully or partially) are prone to having worse results in inductive tasks, as partially shown in the experiment section. We agree that we didn't emphasize this point enough and should show their results regardless. We will **add** the first sentences of this paragraph to our paper. We will also **remove/clarify** the expression **SOTA** and ambiguous bolding in tables of experimental results.

Table 1: **Evaluation on harder dataset: WN18RR** Transductive link prediction results on WN18RR

WN18RR	MRR	Hits@1	Hits@3	Hits@10
ConvE	0.43	0.401	0.44	0.52
ConvR	0.475	0.443	0.489	0.537
RotatE	0.476	0.428	0.492	0.571
TuckER	0.470	0.443	0.482	0.526
ComplEx-N3	0.47	-	-	0.54
NTP2.0* (DistMult)	0.43	-	-	0.49
MINERVA	0.448	0.413	0.456	0.513
Multi-Hop [6]	0.472	0.437	-	0.542
Neural LP	0.435	0.371	0.434	0.566
DRUM, $T = 1$	0.517	0.349	0.594	0.956
DRUM, $T = 2$	0.435	0.370	0.435	0.568
DRUM, $T = 3$	0.486	0.425	0.513	0.586

* Results for DistMult, in [2] authors claim NTP 2.0 is on par with a model similar to DistMult.

Due to lack of space we briefly address other comments in their order of appearance.

Reviewer 1: We really appreciate your thoughtful and detailed comments. Please find in the following our responses.

Matrix Multiplication Idea: Thanks for pointing out [5], we will cite the paper as one of the early works starting the field and write a brief description. *Comparing with NTP, NTP 2.0, dILP:* We will add Table 2, and will write a more detailed explanation about NTP(-λ) and NTP 2.0 because of their importance. However since NTP(-λ) are not scalable to WordNet or FreeBase, we could not present results on larger datasets. NTP 2.0 does not provide results on any large datasets, they claim to be on par with a model similar to distmult which we have added. Thanks for suggesting dILP [7], we will include it in the references. However, unlike our method [7] requires negative examples which is hard to obtain under OWA of modern KGs. Also, [7] is memory-expensive as authors admit, and cannot scale to the size of large KGs (we did not find a publicly available implementation or results on our benchmarks for dILP). *Comparison to relevant work*. please look at the main comment above and we will add a comprehensive comparison table in the appendix. We will clarify/remove SOTA statements. *Harder data-set evaluation* We will also compare our method's performance on WN18RR with that of competitors as in Table 1.

Table 2: Comparison with other reasoning methods. Will be appended to Table 2 of paper.

Datasets	UMLS				Kinship			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
ConvE	0.94	0.92	0.96	0.99	0.83	0.98	0.92	0.98
ComplEx	0.89	0.82	0.96	1	0.81	0.7	0.89	0.98
MINERVA	0.82	0.73	0.90	0.97	0.72	0.60	0.81	0.92
NTP[1]	0.88	0.82	0.92	0.97	0.6	0.48	0.7	0.78
NTP-λ[1]	0.93	0.87	0.98	1	0.8	0.76	0.82	0.89
NTP 2.0	0.76	0.68	0.81	0.88	0.65	0.57	0.69	0.81
DRUM	0.81	0.67	0.94	0.98	0.61	0.46	0.71	0.91

Reviewer 2: We sincerely appreciate your positive feedback and recognition of the significance of our work.

More recent work: We added TuckER, RotatE, and Complex-N3 results for WN18RR. We will also add all of the other suggested methods to a table in the appendix and add them to the references. *Reinitialize embeddings randomly*: This is a great idea, since NTP [1] is not scalable and NTP2.0 [2] doesn't provide public code we have to leave this to future work. *Connections to random walk for KB population literature*: We will summarize these methods and show connections to DRUM in the final paper. *L66*: You're correct, we modified it in the paper. *L125*: Yes, it is the set of relations, we defined $\mathcal{R}$ on line 62. *L189*: this is a very thoughtful comment, we tried a shared RNN but the results were not as good. We believe a single RNN lacks generalizability. *L271*: We asked undergrad CS students. They are not any of the authors or beneficiaries. *L292*: We considered the OWA of KBs and the effect of wrong negative samples (actually true but missing) on generating possible "wrong" rules. Though trivial, the cost function and model need important modifications. Since other methods don't incorporate NS we thought it might not be straight forward.

Reviewer 4: We are really thankful for your insightful comments and positive feedback about our work.

Better notations: Thanks for the suggestion, we will add more explanation about the notation we used. *Comparing regardless of the results:* The results of Multi-Hop [6] and MINERVA [4] are given in Table 1, we will add a comprehensive comparison table to the appendix as well. *WN18RR* We agree that we should have included the result, we will add that to the paper. *hits@1,3* We will add them to the paper, the results for DRUM are about 1 percent better than NeuralLP and for the TransE all the values are very close to zero. *Equation numbers* we agree that it helps the readers, we will add them.

References: [1] End-to-End Differentiable Proving; [2] Towards Neural Theorem Proving at Scale (NTP 2.0); [3] Traversing Knowledge Graphs in Vector Space; [4] Go for a Walk and Arrive at the Answer; [5] Traversing KGs in Vector Space; [6] Multi-Hop Knowledge Graph Reasoning with Reward Shaping; [7] Learning Explanatory Rules from Noisy Data

[1] e.g., On the Kinship dataset DRUM takes 1.2 minutes to run vs +8 hours for NTP(-λ) on the same machine.