We thank all reviewers for their constructive comments. We appreciate that reviewers find the proposed out-of-graph (OOG) link prediction problem to be important as well as novel [R1, R2, R3, R4], the paper well written [R1, R2, R3] and experimental results good [R2, R3, R4]. Due to the page limit, we address the major comments from the reviewers:

Common Comments: Novelty over existing works on few-shot link prediction or GNNs [R1, R4]. We want to emphasize and clarify that our main contribution is neither proposing a general few-shot link prediction method nor a new GNN architecture for general purpose. As clearly stated in the introduction (Line 71-78), our contributions are as follows: 1) the proposal of the few-shot link prediction for **unseen entities** (seen-to-unseen, and unseen-to-unseen), 2) the **transductive meta-learning** framework to solve it by simulating the unseen entities with seen entities during meta-training. Thus any existing GNN models can be trained in our meta-learning framework (Table A).

Experiments against more baselines [R1, R4]. [Chen et al. 19] and [Zhang et al. 20] tackle the prediction of unseen relations of **seen entities**, while our problem deals with **unseen entities**. Yet, we compared against the proposed baselines, MetaR [Chen et al. 19] and FSRL [Zhang et al. 20], on the 3-shot OOG link prediction task. The results in Table A show that they achieve extremely low performance compared to our GENs (Rows 2-3). Further, we trained the baselines in **our meta-learning framework** and obtained significantly improved results (Table A, Rows 4-6). However, their performances are still substantially lower than GENs, which show that GENs' dedicated embedding layers for seen-to-unseen and unseen-to-unseen link prediction are more effective for OOG link prediction. We will include Table A in the revision.

Table A: OOG link prediction results with more baselines. * denotes baselines trained with our meta-learning framework.

	FB15k-237				NELL-995			
	MRR	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**
GMatching [51]	.105	.061	.112	.183	.079	.059	.097	.106
MetaR (Chen et al. 19)	.084	.041	.093	.164	.096	.060	.115	.166
FSRL (Zhang et al. 20)	.090	.058	.096	.150	.085	.064	.095	.126
Ours GMatching*	.238	.168	.263	.372	.139	.092	.151	.235
Ours MetaR*	.316	.235	.341	.492	.213	.145	.247	.352
Ours FSRL*	.259	.186	.281	.404	.161	.106	.181	.275
Ours I-GEN	**.367**	**.281**	**.407**	**.537**	**.285**	**.214**	**.322**	**.426**
Ours T-GEN	**.382**	**.289**	**.430**	**.565**	**.291**	**.217**	**.333**	**.433**

Reviewer #1: Small datasets. FB15k-237 and NELL-995 are large and contain 14,514 and 75,492 nodes respectively.

Reviewer #2: Advantage of a meta-learning framework against retraining from scratch. Our meta-learning framework enables to embed unseen entities without additional re-training which is efficient, and generalizes well to unseen entities. We additionally compared GENs against models trained from scratch (Table B, top two rows), which GENs largely outperform with a fraction of time required to embed unseen entities (Table B). MetaR trained in our meta-learning framework is slower since it uses additional gradient information.

Table B: Retraining.

	MRR	**Time**
DistMult	.094	158.25 sec
TransE	.120	185.09 sec
MetaR*	.316	13.97 sec
I-GEN	.367	**0.99 sec**
T-GEN	**.382**	1.12 sec

Evaluation by the protocol in Sun et al. 20. We found that the performance of our model remains consistent across both Top and Bottom evaluation protocols proposed in Sun et al. 20, and exactly the same as the reported performance.

DistMult initialization. We initialize the pre-trained embedding of seen entities and relations from DistMult for efficient training. However, we also report the results with random initialization in Table 2 of the supplementary file.

Reviewer #3: The distribution of unseen data might not be the same as the seen data. While the initial distribution of unseen data might not be the same as the seen data, when we want to infer relationships among entities, unseen entities should be closer to embeddings of seen entities. When looking at the performance results in Table 2. and the embedding results in Figure A, it can be seen that the performance is good when the unseen data is well aligned with the seen data.

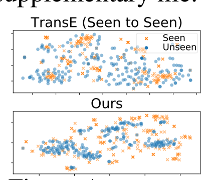

Figure A: T-SNE visualization.

GENs result in better embeddings over seen-to-seen training using TransE. The visualization of the TransE embeddings (Figure A) shows that the embeddings for unseen entities trained in a seen-to-seen manner are not aligned with seen entities, while GEN aligns the unseen entities with the seen entities.

Reviewer #4: Does the meta-training set includes unseen entities? The meta-training set **does not** include real unseen entities at meta-test time, and we simulate the unseen entities with a subset of seen entities during meta-training. We will use the term "simulated unseen" and "real unseen" for further clarification.

How to divide the support set and the query set. We randomly sample the K-triplets associated with each entity for a support set at every episode, and the remaining samples are used as a query set (Line 172-177) in meta-training.

Seen to Seen and Seen to Unseen in Table2? They denote the baseline types. The seen-to-seen are baselines that only handle seen entities, and seen-to-unseen baselines are ones that can tackle seen-to-unseen link prediction tasks.

Denote that the T-GEN utilizes more data during evaluation. T-GEN does not utilize more data, since it performs unseen-to-unseen link prediction for exactly the same set of entities given to all methods. We will clarify this.

No improvements with larger shots (Figure 5). This behavior is consistent with the baselines, and is due to larger shots introducing more noise from weakly-related neighboring nodes.

Can the proposed method improve the benchmark performance? They do, but the ratio of the seen-to-unseen triplets is very small. For example, WN18RR dataset has only 16 seen-to-unseen triplets (0.02%) to evaluate on a query set. Thus, we compared GENs only against seen-to-unseen triplets for WN18RR dataset (Table C). The results demonstrate that the seen-to-unseen performance of the benchmark datasets can also be improved using our GEN.

Table C: Results on WN18RR.

	MRR	**H@1**
DistMult	.000	.000
TransE	.011	.000
MetaR*	.066	.063
I-GEN	**.125**	**.125**