LowFER: Low-rank Bilinear Pooling for Link Prediction — Appendix

A. Proofs

A.1. Proposition 1

Proof. First, we will prove the case for $k = d_e$, with the proof for the case $k = d_r$ following a similar argument. For both cases, we represent entity embedding vector as $\mathbf{e}_i \in \{0,1\}^{|\mathcal{E}|}$, such that only i-th element is 1, and similarly, relation embedding vector as $\mathbf{r}_j \in \{0,1\}^{|\mathcal{R}|}$, such that only j-th element is 1. We represent with $\mathbf{U} \in \mathbb{R}^{d_e \times k d_e}$ and $\mathbf{V} \in \mathbb{R}^{d_r \times k d_e}$ the model parameters, then, given any triple $(e_i, r_j, e_l) \in \mathcal{T}$ with indices (i, j, l), such that $1 \leq i, l \leq |\mathcal{E}|$ and $1 \leq j \leq |\mathcal{R}|$:

For $k = d_e$: We let $\mathbf{U}_{mn} = 1$ for $n = m + (o-1)d_e$, for all m in $\{1, ..., d_e\}$ and for all o in $\{1, ..., k\}$ and 0 otherwise. Further, let $\mathbf{V}_{pq} = 1$ for $p = j$ and $q = (l-1)d_e + i$ and 0 otherwise. Applying $\mathbf{g}(e_i, r_j)$ and taking dot product of the resultant vector with $\mathbf{e}_l$ (Eq. 5) perfectly represents the ground truth as 1. Also, for any triple in $\mathcal{T}'$, a score of 0 is assigned.

For $k = d_r$: We let $\mathbf{U}_{mn} = 1$ for $m = i$ and $n = (l-1)d_e + j$ and 0 otherwise. Further, let $\mathbf{V}_{pq} = 1$ for $q = p + (o-1)d_e$, for all p in $\{1, ..., d_r\}$ and for all o in $\{1, .., k\}$ and 0 otherwise. Rest of the argument follows the same as for $k = d_e$. □

A.2. Proposition 2

Proof. From Eq. 7 and 8, observe that the m-th slice of the core tensor $\mathcal{W}$ on object dimension is approximated by adding k rank-1 matrices, each of which is a cross product between m-th column in $\mathbf{W}_U^{(l)}$ and m-th column in $\mathbf{W}_V^{(l)}$, for all l in $\{1, ..., d_e\}$. Each slice of the core tensor $\mathcal{W}$ on object dimension has a maximum rank $\min(d_e, d_r)$ and from Singular Value Decomposition (SVD), there exists n ($\leq \min(d_e, d_r)$) scaled left singular and scaled right singular vectors whose sum of the cross products is equal to the slice. By choosing these scaled left singular vectors, scaled right singular vectors and zero vectors (in case the rank of the corresponding slice is less than the maximum rank of any such slice) as columns for matrices $\mathbf{W}_U^{(l)}$, $\mathbf{W}_V^{(l)}$, for all l in $\{1, ..., d_e\}$, the core tensor $\mathcal{W}$ is obtained from Eq. 7 with $k \leq \min(d_e, d_r)$. □

Please note that the bounds presented in Table 1 are weak and in general, not very useful. They are derived only for checking the full expressibility of a model, which is also referred to as model being *universal* in Wang et al. (2018), to handle *all-types* of relations with zero error, i.e., perfect reconstruction of the binary tensor $\mathbf{T}$ for a given $\mathcal{KG}$. Since factorization based methods can be seen as approximating the true binary tensor, more useful bounds can be derived by studying the quality of the approximations for a given accuracy level. The bounds for RESCAL, ComplEx and HolE are reported from Wang et al. (2018) while for SimplE (Kazemi & Poole, 2018) and TuckER (Balažević et al., 2019a), from their respective papers.

As discussed in section 4.1, it was first shown in Wang et al. (2018) that TransE is not universal, which was later generalized to other translational models by Kazemi & Poole (2018). RotatE (Sun et al., 2019), a state-of-the-art dissimilarity based model, alleviates the issues of TransE by learning counterclockwise rotations in the complex space. For a triple (h, r, t), RotatE models the tail entity as $\mathbf{t} = \mathbf{h} \circ \mathbf{r}$, where $\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$ are head and tail embeddings and $\mathbf{r} \in \mathbb{C}^d$ is the relation embedding with a restriction on the element-wise modulus, $|r_i| = 1$. Therefore, it only affects the phases of the entity embeddings in the complex vector space. Sun et al. (2019) showed that it can learn *symmetric*, *assymmetric*, *inverse* and *composition* relations (cf. Lemma 1, 2, 3) and degenerates to TransE (cf. Theorem 4). However, we note that RotatE is also not *fully expressive* due to its inability to model the transitive relations in the general case, i.e., irrespective of the size of embedding dimension.

Proposition 3. *RotatE is not fully expressive due to a limitation on the transitive relations.*

Proof. Consider $\{e_1, e_2, e_3\} = \Delta \subset \mathcal{E}$ and $r \in \mathcal{R}$ be a transitive relation on Δ such that $r(e_1, e_2), r(e_2, e_3)$ and $r(e_1, e_3)$ belong to the ground truth. Let $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{r} \in \mathbb{C}^d$ be the embedding vectors for RotatE. Let us assume that $r(e_1, e_2)$ and $r(e_2, e_3)$ hold with RotatE, then we get $\mathbf{e}_2 = \mathbf{r} \circ \mathbf{e}_1$ and $\mathbf{e}_3 = \mathbf{r} \circ \mathbf{e}_2$. From definition of transitive relation we know that $r(e_1, e_2) \wedge r(e_2, e_3) \implies r(e_1, e_3)$, here we obtain $\mathbf{e}_3 = \mathbf{r} \circ \mathbf{r} \circ \mathbf{e}_1$. Therefore for $r(e_1, e_3)$ to hold with RotatE, we must have $\mathbf{r} \circ \mathbf{r} = \mathbf{r} \implies \mathbf{r} = \mathbf{1}$, which in turn suggest $\mathbf{e}_1 = \mathbf{e}_2 = \mathbf{e}_3$ but e_1, e_2, e_3 are distinct entities. More concretely, this condition requires that for all elements of relation embedding r_i, $\cos(\theta_{r,i}) + i\sin(\theta_{r,i})$ should match $\cos(2\theta_{r,i}) + i\sin(2\theta_{r,i})$, which is only possible when $\theta_{r,i} \in \{0, 2\pi\}$, effectively no rotation. □

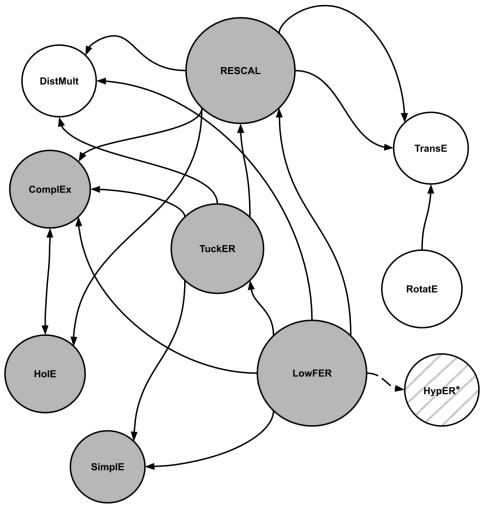

Figure A.1. Subsumption map of KGC models for known relationships: Each node represents a model, where a directed edge shows that the parent node has shown to subsume the child under some conditions. The dotted line shows that the relation is not general enough, where the grey nodes represent *fully expressive* models, the white nodes represent the models that have shown to be not *fully expressive* and dashed ones where this property is not known. The size of a node is relative to the number of outgoing edges. * HypER (Balažević et al., 2019b) has shown to be related to factorization based methods up to a non-linearity, but the authors did not specify any explicit modeling subsumption of other models.

A.3. KGC Scoring Subsumption

In sections 4.2, 4.3 and 4.4 we presented LowFER's relations to other models. In this section, we breifly summarize the subsumption findings of related works. Please note that we only discuss the published findings and refrain from any implied results.

First, Hayashi & Shimbo (2017) showed the equivalence between ComplEx and HolE up to a constant factor using Parseval's theorem[1], which was also discussed in Trouillon & Nickel (2017). Then, the key contributions came from the work of Wang et al. (2018), who showed that RESCAL subsumes TransE, ComplEx, HolE and DistMult by the arguments of ranking tensor. Kazemi & Poole (2018) presented a unified understanding of RESCAL, DistMult, ComplEx and SimplE as *family of bilinear models* under different constraints on the bilinear map. In contrast to the black box 2D-convolution based ConvE model, HypER (Balažević et al., 2019b) showed that 1D-convolution with *hypernetworks* (Ha et al., 2017) come close to well established factorization based methods up to a non-linearity. Balažević et al. (2019a) showed that with certain constraints on the core tensor of the Tucker decomposition (Tucker, 1966), it can subsume the *family of bilinear models*. In this work, we showed that LowFER subsumes TuckER and can be seen as providing low-rank approximation of the core tensor[2] with accurate representation under certain conditions (Proposition 2). We also showed that LowFER can subsume the *family of bilinear models* and HypER up to a non-linearity. Figure A.1[3] provides a network style map for the models discussed here.

[1]For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, it states that $\mathbf{x}^T\mathbf{y} = \frac{1}{d}\mathcal{F}(\mathbf{x})^T\overline{\mathcal{F}(\mathbf{y})}$, where $\mathcal{F} : \mathbb{R}^d \to \mathbb{C}^d$ is the discrete Fourier transform (DFT).

Table A.1. Datasets used for link prediction experiments, where n_e=number of entities, n_r=number of relations and the entities-to-relations ratio n_e/n_r is approximated to the nearest integer.

Dataset	n_e	n_r	n_e/n_r	Training	Validation	Testing
WN18	40,943	18	2275	141,442	5,000	5,000
WN18RR	40,943	11	3722	86,835	3,034	3,134
FB15k	14,951	1,345	11	483,142	50,000	59,071
FB15k-237	14,541	237	61	272,115	17,535	20,466

B. Experiments

In this section, we will present the details of the datasets, evaluation metrics, model implementation, the choice of hyperparameters and report additional experiments.

B.1. Data

We conducted the experiments on four benchmark datasets: WN18 (Bordes et al., 2013) - a subset of Wordnet, WN18RR (Dettmers et al., 2018) - a subset of WN18 created through the removal of inverse relations from validation and test sets, FB15k (Bordes et al., 2013) - a subset of Freebase, and FB15k-237 (Toutanova et al., 2015) - a subset of FB15k created through the removal of inverse relations from validation and test sets. Table A.1 shows the statistics of all the datasets.

B.2. Evaluation Metrics

We report the standard metrics of Mean Reciprocal Rank (MRR) and Hits@k for $k \in \{1, 3, 10\}$. For each test triple (e_s, r, e_o), we score all the triples (e_s, r, e) for all $e \in \mathcal{E}$. We then compute the inverse rank of true triple and average them over all examples. However, Bordes et al. (2013) identified an issue with this evaluation and introduced *filtered* MRR,

[2]The rank of a tensor is the minimal number of rank-1 tensors that yield it in a linear combination. It is known that the tensor rank is NP-hard to compute, and for a 3rd-order tensor $n \times m \times k$, it can be more than $\min(n, m, k)$ but no more than $\min(nm, nk, mk)$ (Miettinen, 2011). Whereas, the n-rank of a tensor $\mathcal{W}$ is the dimension of the vector space spanned by the n-mode vectors, which are the columns of the matrix unfolding $\mathbf{W}_{(n)}$ (De Lathauwer et al., 2000).

[3]https://bit.ly/3k641Ba

Table A.2. Best performing hyper-parameter values for LowFER, where lr=learning rate, dr=decay rate, d_e=entity embedding dimension, d_r=relation embedding dimension, k=LowFER factorization rank, dE=entity embedding dropout, dMFB=MFB dropout, dOut=output dropout and ls=label smoothing. Please note that dE, dMFB and dOut are the same as d#1, d#2 and d#3 as in TuckER (see Appendix A in Balažević et al. (2019a)) respectively.

Dataset	lr	dr	d_e	d_r	k	dE	dMFB	dOut	ls
WN18	0.005	0.995	200	30	10	0.2	0.1	0.2	0.1
WN18RR	0.01	1.0	200	30	30	0.2	0.2	0.3	0.1
FB15k	0.003	0.99	300	30	50	0.2	0.2	0.3	0.0
FB15k-237	0.0005	1.0	200	200	100	0.3	0.4	0.5	0.1

where we only consider triples of the form $\{(e_s, r, e) \mid \forall e \in \mathcal{E}$ s.t. $(e_s, r, e) \notin$ train $\cup$ valid $\cup$ test$\}$ during evaluation. We therefore reported *filtered* MRR for all the experiments. The Hits@k metric computes the percentage of test triples whose ranking is less than or equal to k.

B.3. Implementation and Hyperparameters

We implemented LowFER[4] using the open-source code released by TuckER (Balažević et al., 2019a)[5]. We did random search over the embedding dimensions in $\{30, 50, 100, 200, 300\}$ for d_e and d_r. Further, we varied the factorization rank k in $\{1, 5, 10, 30, 50, 100, 150, 200\}$, with $k = 1$ (LowFER-1) and $k = 10$ (LowFER-10) as baselines. For WN18RR and WN18, we found best $d_e = 200$ and $d_r = 30$ with k value of 30 and 10 respectively. For FB15k-237, we found best $d_e = d_r = 200$ at $k = 100$. All of these embedding dimensions match the best reported in TuckER (Balažević et al., 2019a). However, for FB15k, we found using the configuration of $d_e = 300$ and $d_r = 30$ to be consistently better than $d_e = d_r = 200$. For fair comparison, we also reported the results for $d_e = d_r = 200$ and the best configuration when $d_e = 200$ and $(d_r, k) \leq 200$ (Table 5).

Similar to Balažević et al. (2019a), we used Batch Normalization (Ioffe & Szegedy, 2015) but additionally power normalization and l_2-normalization to stabilize training from large outputs following the Hadamard product in main scoring function (Yu et al., 2017)[6]. We tested the best reported hyperparmeters of Balažević et al. (2019a) with random search and observed good performance in initial testing. With d_e, d_r and k selected, we used fixed set of values for rest of the hyperparameters reported in Balažević et al. (2019a), including learning rate, decay rate, entity embedding dropout, MFB dropout, output dropout and label smoothing (Szegedy et al., 2016; Pereyra et al., 2017) (see Table A.2 for the best hyperparameters). We used Adam (Kingma & Ba, 2015) for optimization. In all the experiments, we trained the models for 500 epochs with batch size 128 and reported the final results on test set.

[4] https://github.com/suamin/LowFER

[5] https://github.com/ibalazevic/TuckER

[6] We observed no performance degradation by removing these additional normalization techniques but we used it in all the experiments to be consistent with prior work of Yu et al. (2017).

Table A.3. Link prediction results on YAGO3-10. Results for DistMult, ComplEx and ConvE are taken from Dettmers et al. (2018) and for RotatE (Sun et al., 2019) (with self-adversarial negative sampling) and HypER (Balažević et al., 2019b) are taken from respective papers.

Model	MRR	Hits@1	Hits@3	Hits@10
DistMult	0.340	0.240	0.380	0.540
ComplEx	0.360	0.260	0.400	0.550
ConvE	0.440	0.350	0.490	0.620
RotatE	0.495	0.402	0.550	0.670
HypER	0.533	0.455	0.580	0.678
LowFER-k*	**0.537**	**0.457**	**0.583**	**0.688**

Table A.4. Link prediction results with LowFER-k* and additional `tanh` non-linearity. The $\downarrow$ shows that the performance went down compared to the linear counterparts reported in Table 2.

Dataset	MRR	Hits@1	Hits@3	Hits@10
FB15k-237 $\downarrow$	0.345	0.256	0.378	0.526
FB15k $\downarrow$	0.818	0.771	0.850	0.898
WN18RR $\downarrow$	0.457	0.429	0.469	0.511
WN18	0.950	0.946	0.952	0.957

B.4. Results on YAGO3-10

We report additional results on YAGO3-10, which is a subset of YAGO3 (Mahdisoltani et al., 2013), consisting of 123,182 entities and 37 relations such that have each entity has at least 10 relations. We used the same best hyperparameters as for WN18RR. Table A.3 shows that our model outperforms state-of-the-art models including RotatE and HypER. It is worth noting that LowFER-k* on YAGO3-10 has only $\sim$26M parameters compared to $\sim$61M parameters of RotatE (Sun et al., 2019), which also includes their self-adversarial negative sampling.

B.5. LowFER with Non-linearity

Similar to Kim et al. (2016), we perform a simple ablation study by adding non-linearity to the LowFER scoring function as follows:

$$\bar{f}(e_s, r, e_o) = (\sigma(\mathbf{S}^k \text{diag}(\mathbf{U}^T \mathbf{e}_s)\mathbf{V}^T \mathbf{r}))^T \mathbf{e}_o$$

where we use hyperbolic tangent $\sigma = \texttt{tanh}$ non-linearity. Applying non-linear activation function can be seen as increasing the representation capacity of the model but Table

A.4 shows that the general performance of LowFER goes down.

B.6. Models Comparison

We compared LowFER with non-linear models including ConvE (Dettmers et al., 2018), R-GCN (Schlichtkrull et al., 2018), Neural LP (Yang et al., 2017), RotatE (Sun et al., 2019)[7], TransE (Bordes et al., 2013), TorusE (Ebisu & Ichise, 2018) and HypER (Balažević et al., 2019b). In linear models, we compared against DistMult (Yang et al., 2015), HolE (Nickel et al., 2016), ComplEx (Trouillon et al., 2016), ANALOGY (Liu et al., 2017), SimplE (Kazemi & Poole, 2018) and state-of-the-art TuckER (Balažević et al., 2019a) model. Results for the Canonical Tensor Decomposition (Lacroix et al., 2018) were not included due to the uncommon choice of extremely large embedding dimensions of $d_e = d_r = 2000$.

Additional models that were not reported in the main results (Table 2) due to partial results but were still outperformed by LowFER include M-Walk (Shen et al., 2018) with their reported metrics of MRR=0.437, Hits@1=0.414 and Hits@3=0.445 on WN18RR and MINERVA (Das et al., 2017) with Hits@10=0.456 on FB15k-237. The results in Table 2 for all the models were taken from Balažević et al. (2019b) and Balažević et al. (2019a). Lastly, in the section 5.4, to perform per relations comparisons, we trained the TuckER models with the best reported configurations in Balažević et al. (2019a) for WN18 and WN18RR.

[7]Where we reported their results in Table 2 without the self-adversarial negative sampling. For fair comparison, see Appendix H in their paper.