Overlapping Spaces for Compact Graph Representations: Supplementary

Anonymous Author(s)
Affiliation
Address
email

A Experimental Setup

A.1 Training Details

All models discussed in Section 5.2 were trained with 2000 iterations. If more than one learning rate was used for a certain dataset (due to problems with the convergence of individual models), all the spaces were evaluated for all learning rates, and the best result was reported for each space. For distortion, the learning rate was 0.1 for all datasets except UCSA312 (Cities), where we had 0.1 and 0.01. For mAP, the learning rate 0.1 was used for all datasets except UCSA312 and CSPhDs, where we had 0.01 and 0.05 for both datasets.

For the experiments in Section 5.3 , we used 5000 iterations for short embeddings and 1000 for long ones (long embeddings converged faster). Hard-negative mining was not used for DSSM training. Instead, large batches of 4096 random training examples (almost 1% of the entire dataset) were used. During the learning process, only the training queries and documents were used. For evaluation, the nearest website was searched among all the documents. The training part was 90% of the dataset, and the quality discrepancy between validation and test sets was quite small. Data samples are given in the table 6.

For the synthetic experiment in Section 5.4 , for all spaces, the learning rates 0.1, 0.05, 0.01, 0.001 were used, and the best result was selected. We had 2000 and 1000 iterations for distortion and mAP, respectively.

A.2 WLA6 Dataset Details

As described in the main text, this dataset is obtained by running the breadth-first search algorithm on the category graph of the English-language Wikipedia (https://en.wikipedia.org/wiki/Special:CategoryTree), starting from the vertex (category) “Linear algebra” and limited to the depth 6 (Wikipedia Linear Algebra 6). We provide this graph along with the texts (names) of the vertices (categories). The resulting graph is very close to being a tree, although there are some cycles. Predictably, hyperbolic space gives a significant profit for this graph, while using product spaces gives almost no additional advantage. The purpose of using this dataset is to check our conclusions on data other than those used in [1] and to evaluate overlapping spaces on a dataset where product spaces do not provide quality gains. Figure 1 visualizes the obtained graph. Table 1 shows full version of the results.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

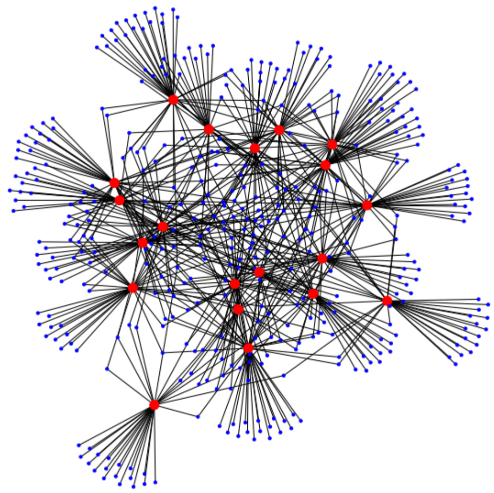

Figure 1: Graph visualization. Red (big) nodes belong to the smaller part.

Table 1: Bipartite graph reconstruction (full version)

	mAP	**distortion**
E_{10}	0.777	0.094
H_{10}	0.794	0.095
S_{10}	0.796	0.096
H_5^2	0.799	0.090
S_5^2	0.796	0.094
$H_5 \times S_5$	0.798	0.090
H_2^5	0.761	0.086
S_2^5	0.773	0.092
$H_2^2 \times E_2 \times S_2^2$	0.796	0.089
$O_{l1}, t = 0$	0.824	0.094
$O_{l1}, t = 1$	0.803	0.082
$O_{l2}, t = 1$	0.814	0.092
best metric space	0.824	0.082
$c -$ dot	**0.863**	**0.079**
$c -$ wips	1	0.091
$O_{mix-l1}, t = 1$	0.986	0.083
$O_{mix-l2}, t = 1$	1	0.070

B Additional Experimental Results

B.1 Our Implementation of Product Spaces vs Original One

Table 2 compares our implementation with the results reported in [1]. It should be noted that we have significantly different algorithms with differing numbers of iterations.

The optimal values of distortion obtained with our algorithm (except for the UCSA312 dataset) are comparable and usually better than those reported in [1]. On UCSA312, the obtained distortion is orders of magnitude better, which can be caused by the proper choice of the learning rate (in our experiments on this dataset, this choice significantly affected the results). These results indicate that our solution is a good starting point to compare different spaces and similarities.

For mAP, we optimize the proxy-loss, in contrast to the canonical implementation, where both metrics were specified for models trained with distortion. Clearly, the results are more stable for our approach:

Table 2: Graph reconstruction: original product spaces vs our implementation

	UCSA312		**CS PhDs**		**Power**		**Facebook**	
	Canon.	Our	Canon.	Our	Canon.	Our	Canon.	Our
				Distortion				
E_{10}	0.0735	0.0032	0.0543	0.0475	0.0917	0.0408	0.0653	0.0487
H_{10}	0.0932	0.0111	0.0502	0.0443	0.0388	0.0348	0.0596	0.0483
S_{10}	0.0598	0.0095	0.0569	0.0503	0.0500	0.0450	0.0661	0.0540
$H_5 \times H_5$	0.0756	0.0057	0.0382	0.0345	0.0365	0.0255	0.0430	0.0372
$S_5 \times S_5$	0.0593	0.0079	0.0579	0.0492	0.0471	0.0433	0.0658	0.0511
$H_5 \times S_5$	0.0622	0.0068	0.0509	0.0337	0.0323	0.0249	0.0402	0.0318
H_2^5	0.0687	0.0059	0.0357	0.0344	0.0396	0.0273	0.0525	0.0439
S_2^5	0.0638	0.0072	0.0570	0.0460	0.0483	0.0418	0.0631	0.0489
$H_2^2 \times E_2 \times S_2^2$	0.0765	0.0044	0.0391	0.0345	0.0380	0.0299	0.0474	0.0406
				mAP				
E_{10}		0.9290	0.8691	0.9487	0.8860	0.9380	0.5801	0.7876
H_{10}		0.9173	0.9310	0.9399	0.8442	0.9385	0.7824	0.7997
$S_{10},$		0.9254	0.8329	0.9578	0.7952	0.9436	0.5562	0.7868
$H_5 \times H_5$		0.9247	0.9628	0.9481	0.8605	0.9415	0.7742	0.8084
$S_5 \times S_5$		0.9231	0.7940	0.9662	0.8059	0.9466	0.5728	0.7891
$H_5 \times S_5$		0.9316	0.9141	0.9654	0.8850	0.9467	0.7414	0.8087
H_2^5		0.9364	0.9694	0.9671	0.8739	0.9508	0.7519	0.7979
S_2^5		0.9281	0.8334	0.9714	0.8818	0.9521	0.5808	0.7915
$H_2^2 \times E_2 \times S_2^2$		0.9391	0.8672	0.9611	0.8152	0.9486	0.5951	0.7970

we do not have such a large spread of values for different spaces. We noticed that directly optimizing ranking losses leads to significant improvements.

B.2 Parametrization of Spherical Space

In Tables 2 and 3 of the main text, we used hyperspherical parameterization of spherical subspaces in product spaces since we fixed the number of stored values for each space. Here, in Tables 3 and 4, we present the extended results, where we fix the mathematical dimension of product spaces and use $d+1$ parameters and simple mappings from Section 3.1, equation 4, as done in [1]. We can see that our implementation gives results comparable to the original ones in distortion setup and significantly better for mAP, which is associated with using the proxy-loss instead of distortion.

Table 3: Graph reconstruction with distortion loss, top results are highlighted, metrics only.

Signature	**UCSA312**	**CS PhDs**	**Power**	**Facebook**	**WLA6**
E_{10}	**0.00318**	0.0475	0.0408	0.0487	0.0530
H_{10}	0.01114	0.0443	0.0348	0.0483	**0.0279**
S_{10}	0.00951	0.0503	0.0450	0.0540	0.0589
$H_5^2 \equiv H_5 \times H_5$	0.00573	0.0345	0.0255	0.0372	**0.0279**
$S_5 \times S_5 \equiv S_5^2$	0.00792	0.0492	0.0433	0.0511	0.0585
$H_5 \times S_5$	0.00681	**0.0337**	**0.0249**	**0.0318**	0.0296
H_2^5	0.00592	0.0344	0.0273	0.0439	0.0356
S_2^5	0.00720	0.0460	0.0418	0.0489	0.0549
$H_2^2 \times E_2 \times S_2^2$	0.00436	0.0345	0.0299	0.0406	0.0405
$O_{l1}, t=0$	**0.00356**	0.0368	0.0281	0.0458	0.0286
$O_{l1}, t=1$	**0.00330**	**0.0300**	**0.0231**	**0.0371**	**0.0272**
$O_{l2}, t=1$	0.00530	**0.0328**	**0.0246**	**0.0324**	**0.0278**

Table 5: Comparison of proxy-losses, mAP

	UCSA312			**CS PhD**		
$P \sim$	e^{-d}	$e^{1/d}$	$1/d$	e^{-d}	$e^{1/d}$	$1/d$
E_{10}	0.929	0.911	0.899	0.949	0.956	0.831
H_{10}	0.917	0.807	0.885	0.940	0.749	0.764
S_{10}	0.925	0.797	0.838	0.958	0.572	0.689
H_5^2	0.925	0.890	0.883	0.948	0.976	0.723
S_5^2	0.923	0.802	0.858	0.966	0.748	0.775
$H_5 \times S_5$	0.932	0.838	0.865	0.965	0.804	0.721
H_2^5	0.936	0.896	0.903	0.967	0.998	0.823
S_2^5	0.928	0.856	0.871	0.971	0.876	0.881
$H_2^2 \times E_2 \times S_2^2$	0.939	0.872	0.865	0.961	0.884	0.689
$O_{l1}, t = 0$	0.952	0.933	0.872	0.988	0.961	0.762
$O_{l1}, t = 1$	0.952	0.947	0.877	0.990	0.963	0.815
$O_{l2}, t = 1$	0.952	0.939	0.880	0.994	0.979	0.810
$c -$ dot	1	1	0.777	1	0.999	0.917

Table 4: Graph reconstruction with mAP ranking loss, top results are highlighted, metrics only.

Signature	**UCSA312**	**CS PhDs**	**Power**	**Facebook**	**WLA6**
E_{10}	0.9290	0.9487	0.9380	0.7876	0.7199
H_{10}	0.9173	0.9399	0.9385	0.7997	0.9617
S_{10}	0.9254	0.9578	0.9436	0.7868	0.7287
H_5^2	0.9247	0.9481	0.9415	0.8084	**0.9682**
S_5^2	0.9231	0.9662	0.9466	0.7891	0.7353
$H_5 \times S_5$	0.9316	0.9654	0.9467	0.8087	**0.9779**
H_2^5	0.9364	0.9671	0.9508	0.7979	0.8597
S_2^5	0.9281	0.9714	0.9521	0.7915	0.7346
$H_2^2 \times E_2 \times S_2^2$	0.9391	0.9611	0.9486	0.7970	0.6796
$O_{l1}, t = 0$	**0.9522**	**0.9879**	**0.9728**	**0.8093**	0.6759
$O_{l1}, t = 1$	**0.9522**	**0.9904**	**0.9762**	**0.8185**	0.9598
$O_{l2}, t = 1$	**0.9522**	**0.9938**	**0.9907**	**0.8326**	**0.9694**

B.3 Other Ways of Converting Distances to Probabilities

For the proxy-loss, we additionally experimented with other ways of converting distances to probabilities. Let us write L_{proxy} in the general form:

$$L_{proxy} = -\sum_{(v,u)\in E} \log \mathrm{P}((v,u) \in E) = -\sum_{(v,u)\in E} \log \frac{t\big(d_U(f(v), f(u))\big)}{\sum\limits_{w\in V} t\big(d_U(f(v), f(w))\big)}, \tag{1}$$

where $t(d)$ is a function that decreases with distance d. We compare the following alternatives for $t(d)$:

$$t_1(d) = \exp(-d), t_2(d) = \exp\left(\frac{1}{\min(d, d_0)}\right), t_3(d) = \frac{1}{\min(d, d_0)},$$

where d_0 is a small constant.

Recall that t_1 was used in the main text and it seems to be the most natural choice.[1] Table 5 compares the options and shows that the best results are indeed achieved with t_1.

[1] Note that this is the softmax over the inverted distances.

Table 6: Search query examples

Query	**Web site**
Kris Wallace	en.wikipedia.org/wiki/Chris_Wallace
1980: Mitsubishi produces one million cars...	en.wikipedia.org/wiki/Mitsubishi_Motors
code napoleon	en.wikipedia.org/wiki/Napoleonic_Code

References

[1] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. 2019. Learning mixed-curvature representations in product spaces. *International Conference on Learning Representations (ICLR)* (2019).