Supplementary Material

A Overview

This document is supplementary material for the paper "Rep the Set: Neural Networks for Learning Set Representations". It is organized as follows. We will prove in Section B the Theorem 1. In Section C, we will present the proof of Proposition 1. In Section D, we will give details about the datasets we used in our experiments. Finally, in Section E, we perform a sensitivity analysis, and we present the features that the model learns on a simple synthetic dataset.

B Proof of Theorem 1

For the reader's convenience we will restate Theorem 1 from the article.

Theorem 2. *Let X be a set having elements from a countable or uncountable universe. The proposed architecture is invariant to the permutation of elements in X.*

Proof. Let Π_n be the set of all permutations of the integers from 1 to n. Let $\pi \in \Pi_{|X|}$ be an arbitrary permutation. We will apply π to the input set X. The bipartite matching problem then becomes:

$$\begin{aligned} &\max \sum_{i=1}^{|X|}\sum_{j=1}^{|Y|} z_{ij} f(\mathbf{v}_{\pi(i)}, \mathbf{u}_j) \\ &\text{subject to:} \\ &\sum_{i=1}^{|X|} z_{ij} \leq 1 \quad \forall j \in \{1,\dots,|Y|\} \\ &\sum_{j=1}^{|Y|} z_{ij} \leq 1 \quad \forall i \in \{1,\dots,|X|\} \\ &z_{ij} \geq 0 \quad \forall i \in \{1,\dots,|X|\}, \forall j \in \{1,\dots,|Y|\} \end{aligned} \tag{7}$$

The constraints of the optimization problem remain intact since summing the elements of a set is a permutation invariant function. Moreover, it holds that:

$$\sum_{i=1}^{|X|}\sum_{j=1}^{|Y|} z_{ij} f(\mathbf{v}_{\pi(i)}, \mathbf{u}_j) = \sum_{i=1}^{|X|}\sum_{j=1}^{|Y|} z_{\pi^{-1}(i)j} f(\mathbf{v}_i, \mathbf{u}_j) \tag{8}$$

The sets of variables that lead to the optimal solutions of problems (1) in the main paper and (7) above are identical, i. e., $z^*_{\pi^{-1}(i)j} = z^*_{ij}$, $\forall i \in \{1,\dots,|X|\}, \forall j \in \{1,\dots,|Y|\}$. Hence, the optimal value of the bipartite matching problem is the same for all $n!$ permutations of the input, and therefore, the proposed model maps all permutations of the input into the same representation. □

C Proof of Proposition 1

For the reader's convenience we will restate Proposition 1 from the article.

Proposition 2. *The optimization problem defined in Equation 6 is an upper bound to the bipartite matching problem defined in Equation 1.*

Proof. We assume without loss of generality that $|X| \geq |Y|$. Let $\mathbf{D}^*$ be the optimal solution to the relaxed problem, i. e., $\mathbf{D}^*_{ij} = z_{ij}$. Therefore, it holds that:

$$\mathbf{D}^*_{ij} = \begin{cases} 1 & \text{if } i = \arg\max_k f(\mathbf{v}_k, \mathbf{u}_j) \wedge \max_k f(\mathbf{v}_k, \mathbf{u}_j) > 0 \\ 0 & \text{otherwise} \end{cases}$$

The relaxed problem is allowed to match multiple elements of Y with the same element of X. Specifically, the optimal solution of the relaxed problem matches an element of Y with an element of X if their inner product is positive and is the highest among the inner products between that element of Y and all the elements of X. Then, for every j, let $i^* = \arg\max_k f(\mathbf{v}_k, \mathbf{u}_j)$. For any feasible solution $\mathbf{D}$ of the exact problem, and for any j, we have:

$$\begin{aligned} \sum_{i=1}^{|X|} \mathbf{D}_{ij} f(\mathbf{v}_i, \mathbf{u}_j) &\leq \sum_{i=1}^{|X|} \mathbf{D}_{ij} f(\mathbf{v}_{i^*}, \mathbf{u}_j) \\ &= f(\mathbf{v}_{i^*}, \mathbf{u}_j) \sum_{i=1}^{|X|} \mathbf{D}_{ij} \\ &\leq f(\mathbf{v}_{i^*}, \mathbf{u}_j) \\ &= \sum_{i=1}^{|X|} \mathbf{D}^*_{ij} f(\mathbf{v}_i, \mathbf{u}_j) \end{aligned} \tag{9}$$

Therefore, the objective value of the relaxed problem (obtained by $\mathbf{D}^*$) gives an upper to the exact problem. □

D Datasets

D.1 Text Categorization Datasets

We evaluated all approaches on 8 supervised document datasets: (1) BBCSPORT: BBC sports articles between 2004-2005, (2) TWITTER: a set of tweets labeled with sentiments 'positive', 'negative', or 'neutral' (the set is reduced due to the unavailability of some tweets), (3) RECIPE: a set of recipe procedure descriptions labeled by their region of origin, (4) OHSUMED: a collection of medical abstracts categorized by different cardiovascular disease groups (for computational efficiency we subsample the dataset, using the first 10 classes), (5) CLASSIC: sets of sentences from academic

Table 4: Datasets used in text categorization experiments.

Dataset	n	Voc	Unique Words(avg)	y
BBCSPORT	517	13243	117	5
TWITTER	2176	6344	9.9	3
RECIPE	3059	5708	48.5	15
OHSUMED	3999	31789	59.2	10
CLASSIC	4965	24277	38.6	4
REUTERS	5485	22425	37.1	8
AMAZON	5600	42063	45.0	4
20NG	11293	29671	72	20

papers, labeled by publisher name, (6) REUTERS: a classic news dataset labeled by news topics (we use the 8-class version with train/test split as described in Cachopo (2007), (7) AMAZON: a set of Amazon reviews which are labeled by category product in books, dvd, electronics, kitchen (as opposed to by sentiment), (8) 20NG: news articles classified into 20 different categories (we use the "bydate" train/test split by Cachopo (2007)). We preprocess all datasets by removing all words in the SMART stop word list (Salton and Buckley, 1971). Table 4 shows statistics of the 8 datasets that were used for the evaluation. We obtained a distributed representation for each word from a publicly available set of pre-trained vectors[1]. For datasets that do not come with a predefined train/test split, we report the average accuracy over five 70/30 train/test splits as well as the standard deviation.

D.2 Graph Classification Datasets

We evaluated the proposed architecture on the following 5 datasets: (1) MUTAG, (2) PROTEINS, (3) IMDB-BINARY, (4) IMDB-MULTI, and (5) REDDIT-BINARY.

MUTAG contains mutagenic aromatic and heteroaromatic nitro compounds. Each chemical compound is labeled according to whether or not it has mutagenic effect on the Gram-negative bacterium Salmonella typhimurium (Debnath et al., 1991). PROTEINS consists of proteins represented as graphs where vertices are secondary structure elements and there is an edge between two vertices if they are neighbors in the amino-acid sequence or in 3D space. The task is to classify proteins into enzymes and non-enzymes (Borgwardt et al., 2005). IMDB-BINARY and IMDB-MULTI contain movie collaboration graphs. The vertices of each graph represent actors/actresses and two vertices are connected by an edge if the corresponding actors/actresses appear in the same movie. Each graph is the ego-network of an actor/actress, and the task is to predict which genre an ego-network belongs to (Yanardag and Vishwanathan, 2015). REDDIT-BINARY consists of online discussion threads represented as graphs. Each vertex corresponds to a user, and two users are connected by an edge if one of them responded to at least one of the other's comments. The task is to classify graphs into either communities (Yanardag and Vishwanathan, 2015). A summary of the datasets is given in Table 5.

[1] `https://code.google.com/archive/p/word2vec/`

Table 5: Datasets used in graph classification.

Dataset	#Graphs	y	Nodes(avg)	Edges(avg)
MUTAG	188	2	17.93	19.79
PROTEINS	1113	2	39.06	72.82
IMDB BINARY	1000	2	19.77	96.53
IMDB MULTI	1500	3	13.00	65.94
REDDIT BINARY	2000	2	429.63	497.75

E Experimental Evaluation

E.1 Sensitivity Analysis

The proposed RepSet and ApproxRepSet models involve two main parameters: (1) the number of hidden sets m, and (2) the cardinalities of the hidden sets $|Y_i|, i = 1, \ldots, m$. We next investigate how these two parameters influence the performance of the RepSet model. Specifically, in Figures 5 and 6, we examine how the different choices of these parameters affect the performance of RepSet on the TWITTER and RECIPE datasets, respectively. We measure the test error as a function of the two parameters. Note that each hidden set Y_i can have a different cardinality compared to the other sets. However, we set the cardinalities of all hidden sets to the same value. We observe that on TWITTER, the number of hidden sets m does not have a large impact on the performance, especially for small cardinalities of the hidden sets ($|Y_i| \leq 50$). For most cardinalities, the test error is within 1% to 3% when varying this parameter.

Furthermore, in most cases, the best performance is attained when the number of hidden sets is small ($m \leq 20$). Similar behavior is also observed for the second parameter on the TWITTER dataset. For most values of m, the test error changes only slightly when varying the cardinalities of the hidden sets. For $m \geq 50$, the model produces best results when the cardinalities of the hidden sets $|Y_i|$ are close to 20. On the other hand, for small values of m, the model yields good performance even when the cardinalities of the hidden sets $|Y_i|$ are large. On the RECIPE dataset, both parameters have a higher impact on the performance of the RepSet model. In general, small values

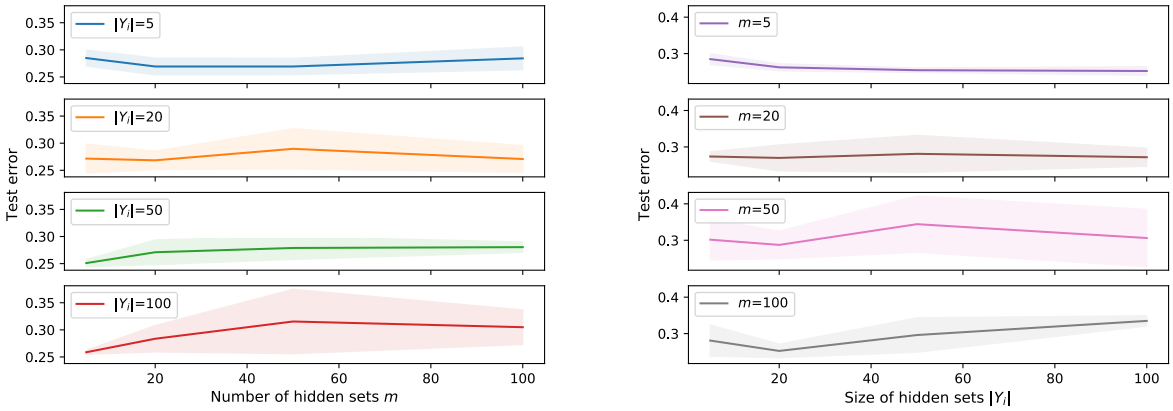

Figure 5: Average test error of the RepSet model with respect to the number of hidden sets m (left) and the size of the hidden sets $|Y_i|$ (right) on the TWITTER dataset.

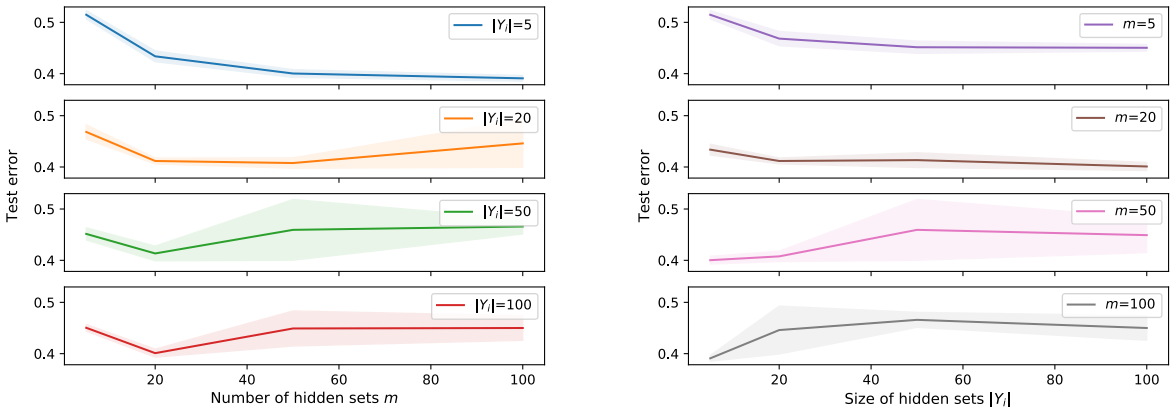

Figure 6: Average test error of the RepSet model with respect to the number of hidden sets m (left) and the size of the hidden sets $|Y_i|$ (right) on the RECIPE dataset.

of m lead to higher test error than larger values of m. For most values of $|Y_i|$, values of m between 20 and 50 result in the lowest test error. As regards the size of the hidden sets $|Y_i|$, there is no consistency in the obtained results. Specifically, for small values of m ($m \leq 20$), large cardinalities of the hidden sets result in better performace, while for large values of m ($m \geq 50$), small cardinalities lead to smaller error.

E.2 Synthetic Data

We first demonstrate the proposed RepSet architecture on a very simple dataset. The dataset consists of 4 sets of 2-dimensional vectors. The cardinality of all sets is equal to 2. The elements of the 4 sets are illustrated in Figure 7. Although seemingly simple, this dataset may prove challenging for several algorithms that apply aggregation mechanisms to the elements of the sets since all 4 sets have identical centroids while the sum of their elements is also the same for all of them. To learn to classify these sets, we used an instance of the proposed model consisting of 2 hidden sets of cardinality equal to 2. The model managed easily to discriminate between the 4 sets and to achieve perfect accuracy. A question that arises at this point is what kind of features the hidden sets of the model learn during training. Hence, besides the input sets, Figure 7 also shows the vectors of the 2 hidden sets. The hidden sets learned very similar patterns, which indicates that less than 2 hidden sets may be required. In fact, we observed that even with one hidden set, the model can achieve perfect performance.

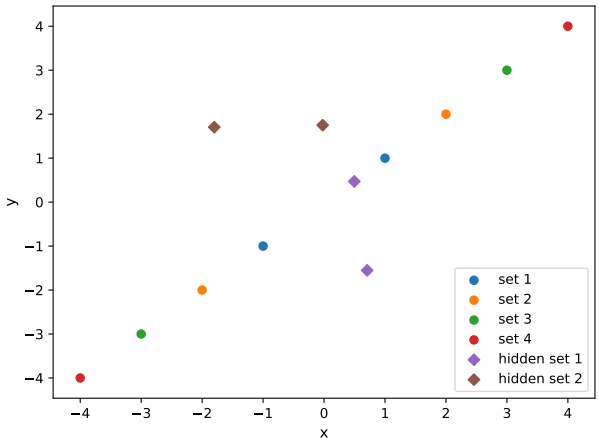

Figure 7: A very simple dataset consisting of 4 examples (i. e., sets). Each set contains a pair of 2-dimensional vectors (circles). The diamonds indicate the "hidden sets" that the proposed model learned during training.