SET2BOX: Similarity Preserving Representation Learning for Sets (Supplementary Document)

Anonymous Author(s)

1 Dataset Details

We provide some statistis of the real-world datasets used for the experiment in Table 1. Below, We provide the details of each dataset:

- **Yelp** consists of user ratings on locations (e.g., hotels and restaurants), and each set is a group of locations that a user rated. Ratings higher than 3 are considered.
- **Amazon** contains reviews of products (specifically, those categorized as Movies & TV) by the users. In the dataset, each user has at least 5 reviews. A group of products reviewed by the same user is abstracted as a set.
- **Netflix** is a collections of movie ratings from users. Each set is a set of movies rated by each user and each entity is a movie. We consider ratings higher than 3.
- **Gplus** is a directed social network that consists of 'circles' from Google+. Each set is a group of neighborhood nodes of each node.
- **Twitter** also is a directed social networks consisting of 'circles' in Twitter. Each set is a group of neighbors of each node.
- **MovieLens (ML1, ML10, and ML20)** are collections of movie ratings from anonymous users. Each set is a group of movies that a user rated. Ratings higher than 3 are considered.

Table 1: Statistics of the 8 real-world datasets: the number of entities $|\mathcal{E}|$, the number of sets $|\mathcal{S}|$, the maximum set size $\mathbf{max}_{s\in\mathcal{S}}|s|$, and the size of the dataset $\mathbf{sum}_{s\in\mathcal{S}}|s|$.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{S}\rvert$	$\mathbf{max}_{s\in\mathcal{S}}\lvert s\rvert$	$\mathbf{sum}_{s\in\mathcal{S}}\lvert s\rvert$
Yelp (YP)	25,252	25,656	649	467K
Amazon (AM)	55,700	105,655	555	858K
Netflix (NF)	17,769	478,615	12,206	56.92M
Gplus (GP)	107,596	72,271	5,056	13.71M
Twitter (TW)	81,305	70,097	1,205	1.76M
MovieLens 1M (ML1) [2]	3,533	6,038	1,435	575K
MovieLens 10M (ML10) [2]	10,472	69,816	3,375	5.89M
MovieLens 20M (ML20) [2]	22,884	138,362	4,168	12.20M

2 Experimental Details

In this section, we provide the details of the settings of the experiments conducted in the main paper.

Hyperparameter Tuning Table 2 describes the hyperparameter search space of each method. The number of training samples, $|\mathcal{T}^+|$ and $|\mathcal{T}^-|$, are both set to 10 for Set2Box, Set2Box$^+$, and their variants. For the vector-based methods, Set2Vec and Set2Vec$^+$, since three pairwise relations are extractable from each triple, $\lceil \frac{7}{3}|\mathcal{T}^+| \rceil$ positive triples and $\lceil \frac{7}{3}|\mathcal{T}^-| \rceil$ negative samples are used for training. We consistently set batch size to 512 and used Adam optimizer. In Set2Box$^+$, we consistently set the softmax temperature τ to 1.

Table 2: Search space of each method.

Method	Hyperparameter	Selection Pool
Set2Box	Learning rate	0.001, 0.01
	Box smoothing parameter β	1, 2, 4
Set2Box$^+$	Learning rate	0.001, 0.01
	Box smoothing parameter β	1, 2, 4
	Joint training coefficient λ	0, 0.001, 0.01, 0.1, 1

Projected Graph Generation To obtain the entity features for Set2Vec$^+$, we generate a projected graph whose nodes are entities and edges are connected if any two entities have appeared in the same set. Specifically, we generate a weighted graph by assigning weight (specifically, the number of sets the two entities share) to each edge and perform a biased random walk. Based on this, we apply node2vec [1], a popular random walk-based network embedding method, to obtain the feature of each entity.

Encoding Costs We summarize the encoding cost of each method, including the variants of Set2Box and Set2Box$^+$, in Table 3.

Table 3: Encoding cost of each method covered in this work: number of sets $|\mathcal{S}|$, dimension d, number of subspaces D, and number of key boxes (vectors) in each subspace K.

Method	Encoding Cost (bits)
Set2Bin	$d\|\mathcal{S}\|$
Set2Vec	$32d\|\mathcal{S}\|$
Set2Vec$^+$	$32d\|\mathcal{S}\|$
Set2Box-order	$32d\|\mathcal{S}\|$
Set2Box-PQ	$64Kd + 2\|\mathcal{S}\|D\log_2 K$
Set2Box	$64d\|\mathcal{S}\|$
Set2Box$^+$	$64Kd + \|\mathcal{S}\|D\log_2 K$

3 Extra Experimental Results

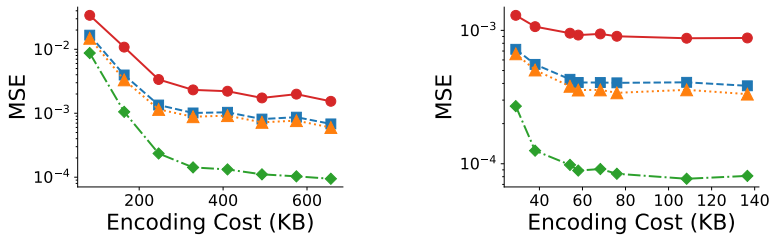
In this section, we provide extra results of the experiments regarding **Q1**, **Q2**, and **Q3**.

Q1. Accuracy & Conciseness: Does Set2Box$^+$ derive concise and accurate set representations than its competitors?

Q2. Effectiveness: How Why Set2Box$^+$ yield concise and accurate set representations? Are the proposed schemes effective?

Q3. Effects of Parameters: How do the parameters of Set2Box$^+$ affect the quality of set representations?

Q1. Accuracy & Conciseness As shown in Figure 1, both SET2BOX and SET2BOX$^+$ benefit from the larger encoding cost.



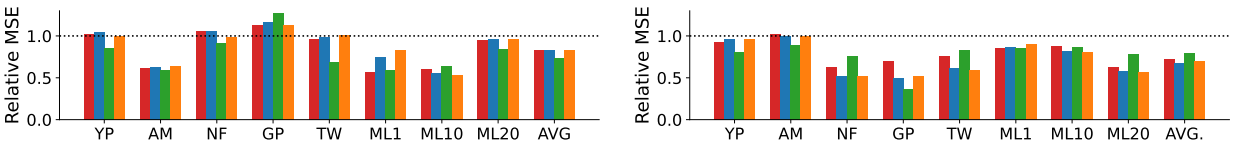
(a) SET2BOX (b) SET2BOX$^+$

Figure 1: SET2BOX and SET2BOX$^+$ gives more accurate estimation of Overlap Coefficient, Cosine Similarity, Jaccard Index, and Dice Index when more encoding cost is used in Yelp.

Q2. Effectiveness To analyze the effectiveness of **(1)** the box quantization and **(2)** the joint training in SET2BOX$^+$, we measure the following relative MSEs:

$$\frac{\text{MSE of SET2BOX-BQ}}{\text{MSE of SET2BOX-PQ}} \quad \text{and} \quad \frac{\text{MSE of SET2BOX}^+}{\text{MSE of SET2BOX-BQ}}$$

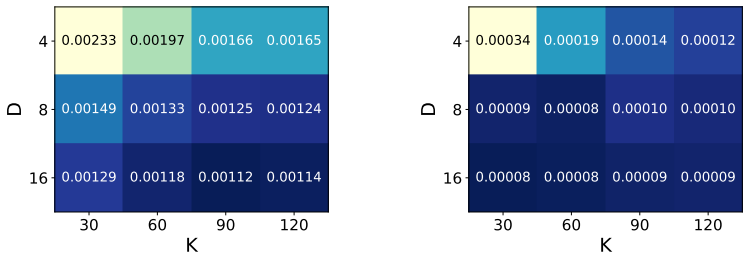
of each dataset. Figure 2a demonstrates that SET2BOX-BQ generally derives more accurate set representations compared to SET2BOX-PQ, implying the effectiveness of the proposed box quantization scheme. Also, Figure 2b shows that SET2BOX$^+$ is superior compared to SET2BOX-BQ in most datasets indicating that jointly training the reconstruction boxes with the original ones leads to accurate boxes.



(a) Effects of the proposed box quantization scheme (b) Effects of joint training scheme

Figure 2: Relative MSEs of Overlap Coefficient, Cosine Similarity, Jaccard Index, and Dice Index in each dataset. The proposed schemes: box quantization (SET2BOX-PQ vs. SET2BOX-BQ) and joint training (SET2BOX-BQ vs. SET2BOX$^+$) improve the accuracy.

Q3. Effects of Parameters We analyze how the parameters in SET2BOX$^+$ affect the accuracy.
∘ **Effects of D & K:** The number of subspaces (D) and the number of key boxes in each subspace (K) are key parameters that controls the encoding cost of SET2BOX$^+$. In Figure 3, we investigate how the performance of SET2BOX$^+$ changes for different D and K values. Typically, the accuracy improves as D and K increases, but so does the encoding cost. In addition, the performance of SET2BOX$^+$ is more heavily affected by D compared to K.



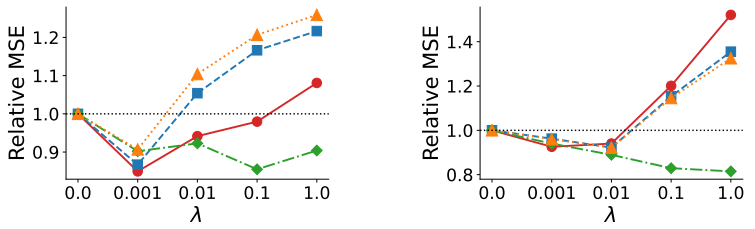
(a) MovieLens1M (b) Yelp

Figure 3: Effects of K and D used in SET2BOX$^+$ in terms of Jaccard Index.

◦ **Effects of λ:** To observe how the coefficient λ in our final objective,

$$\mathcal{L} = \sum_{\{s_i, s_j, s_k\} \in \mathcal{T}} \lambda \left(\mathcal{J}_1 + \mathcal{J}_2 + \mathcal{J}_3 + \mathcal{J}_4 + \mathcal{J}_5 + \mathcal{J}_6 + \mathcal{J}_7 \right) + \mathcal{J}_8,$$

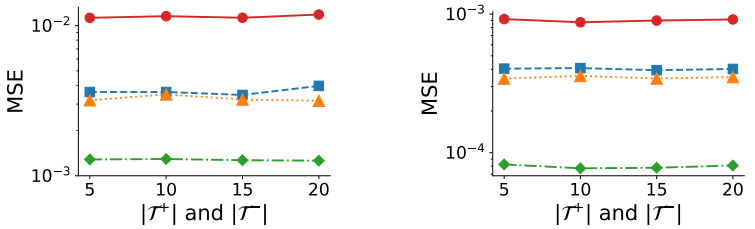
affects the accuracy of SET2BOX$^+$, we measured the relative MSE (based on the MSE when $\lambda = 0$) with respect to different λs. As shown in Figure 4, the accuracy of SET2BOX$^+$ is affected by λ: joint training is beneficial, as demonstrated in the main paper, but overemphasizing the joint views may prevent SET2BOX$^+$ to learn meaningful reconstructed boxes.



(a) MovieLens1M (b) Yelp

Figure 4: Effects of λ in SET2BOX$^+$ regarding Overlap Coefficient, Cosine Similarity, Jaccard Index, and Dice Index.

◦ **Effects of $|\mathcal{T}^+|$ and $|\mathcal{T}^-|$:** In Figure 5, we observe the effects of the number of training samples, $|\mathcal{T}^+|$ and $|\mathcal{T}^-|$, in SET2BOX$^+$. We can see that the accuracy is robust to the parameters, and thus using only a small number of samples for training is enough (we consistently use $|\mathcal{T}^+| = |\mathcal{T}^-| = 10$ in all experiments).



(a) MovieLens1M (b) Yelp

Figure 5: Effects of $|\mathcal{T}^+|$ and $|\mathcal{T}^-|$ in SET2BOX$^+$ regarding Overlap Coefficient, Cosine Similarity, Jaccard Index, and Dice Index.

4 Box Properties

In Table 4, we list six set properties that are supported by boxes.

Table 4: Boxes satisfy various set properties.

Property	**Box Representations**
1. Transitivity Law	$\mathrm{B}_X \subset \mathrm{B}_Y,\ \mathrm{B}_Y \subset \mathrm{B}_Z \rightarrow \mathrm{B}_X \subset \mathrm{B}_Z$
2. Idempotent Law	$\mathrm{B}_X \cup \mathrm{B}_X = \mathrm{B}_X$ $\mathrm{B}_X \cap \mathrm{B}_X = \mathrm{B}_X$
3. Commutative Law	$\mathrm{B}_X \cup \mathrm{B}_Y = \mathrm{B}_Y \cup \mathrm{B}_X$ $\mathrm{B}_X \cap \mathrm{B}_Y = \mathrm{B}_Y \cap \mathrm{B}_X$
4. Associative Law	$\mathrm{B}_X \cup (\mathrm{B}_Y \cup \mathrm{B}_Z) = (\mathrm{B}_X \cup \mathrm{B}_Y) \cup \mathrm{B}_Z$ $\mathrm{B}_X \cap (\mathrm{B}_Y \cap \mathrm{B}_Z) = (\mathrm{B}_X \cap \mathrm{B}_Y) \cap \mathrm{B}_Z$
5. Absorption Law	$\mathrm{B}_X \cup (\mathrm{B}_X \cap \mathrm{B}_Y) = \mathrm{B}_X$ $\mathrm{B}_X \cap (\mathrm{B}_X \cup \mathrm{B}_Y) = \mathrm{B}_X$
6. Distributive Law	$\mathrm{B}_X \cap (\mathrm{B}_Y \cup \mathrm{B}_Z) = (\mathrm{B}_X \cap \mathrm{B}_Y) \cup (\mathrm{B}_X \cap \mathrm{B}_Z)$ $\mathrm{B}_X \cup (\mathrm{B}_Y \cap \mathrm{B}_Z) = (\mathrm{B}_X \cup \mathrm{B}_Y) \cap (\mathrm{B}_X \cup \mathrm{B}_Z)$

References

[1] A. Grover and J. Leskovec, *node2vec: Scalable feature learning for networks*, in KDD, 2016.

[2] F. M. Harper and J. A. Konstan, *The movielens datasets: History and context*, TiiS, 5 (2015), pp. 1–19.