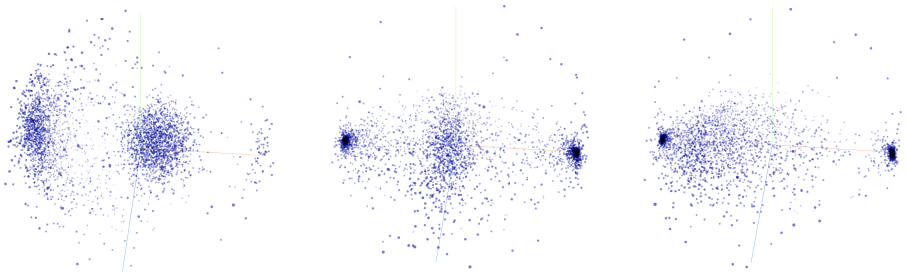

Figure 3: Projecting 50-dimensional embeddings obtained by training a simple neural network without SSE (Left), and with SSE-Graph (Center) , SSE-SE (Right) into 3D space using PCA.

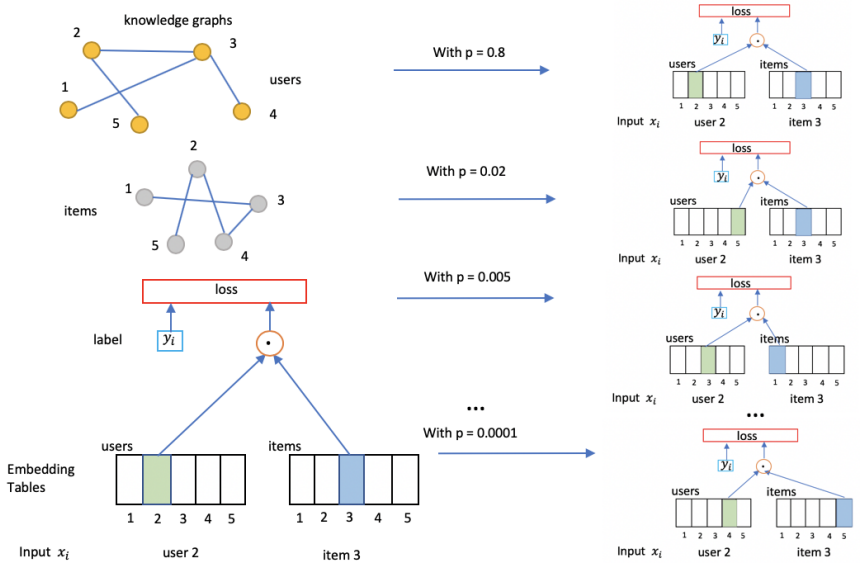

Figure 4: Illustration of how SSE-Graph algorithm in Figure 1 works for a simple neural network.

5 Appendix

We find that SSE with knowledge graphs, i.e., SSE-Graph, can force similar embeddings to cluster when compared to the original neural network without SSE-Graph. In Figure 3, one can easily see that more embeddings tend to cluster into 2 black holes after applying SSE-Graph when embeddings are projected into 3D spaces using PCA. Interestingly, a similar phenomenon occurs when assuming the knowledge graph is a complete graph, which we would introduce as SSE-SE below.

For experiments in Section 4, we use Julia and C++ to implement SGD. For experiments in Section 4, and Section 4, we use Tensorflow and SGD/Adam Optimizer. For experiments in Section 4, we use Pytorch and Adam with noam decay scheme and warm-up. We find that none of these choices affect the strong empirical results supporting the effectiveness of our proposed methods, especially the SSE-SE. In any deep learning frameworks, we can introduce stochasticity to the original embedding look-up behaviors and easily implement SSE-Layer in Figure 1 as a custom operator.

5.1 Neural Networks with One Hidden Layer

To run SSE-Graph, we need to construct good-quality knowledge graphs on embeddings. We managed to match movies in Movielens1m and Movielens10m datasets to IMDB websites, therefore we can extract plentiful information for each movie, such as the cast of the movies, user reviews and so on. For simplicity reason, we construct the knowledge graph on item-side embeddings using the cast of movies. Two items are connected by an edge when they share one or more actors/actresses. For user side, we do not have good quality graphs: we are only able to create a graph on users in Movielens1m dataset based on their age groups but we do not have any side information on users in Movielens10m dataset. When running experiments, we do a parameter sweep for weight decay parameter and then fix it before tuning the parameters for SSE-Graph and SSE-SE. We utilize different ρ and p for user and item embedding tables respectively. The optimal parameters are stated in Table 1 and Table 2. We use the learning rate of 0.01 in all SGD experiments.

In the first leg of experiments, we examine users with fewer than 60 ratings in Movielens1m and Movielens10m datasets. In this scenario, the graph should carry higher importance. One can easily see from Table 1 that without using graph information, our proposed SSE-SE is the best performing matrix factorization algorithms among all methods, including popular ALS-MF and SGD-MF in terms of RMSE. With Graph information, our proposed SSE-Graph is performing significantly better than the Graph Laplacian Regularized Matrix Factorization method. This indicates that our SSE-Graph has great potentials over Graph Laplacian Regularization as we do not explicitly penalize the distances across embeddings but rather we implicitly penalize the effects of similar embeddings on the loss.

In the second leg of experiments, we remove the constraints on the maximum number of ratings per user. We want to show that SSE-SE can be a good alternative when graph information is not available. We follow the same procedures in [31, 32]. In Table 2, we can see that SSE-SE can be used with dropout to achieve the smallest RMSE across Douban, Movielens10m, and Netflix datasets. In Table 3, one can see that SSE-SE is more effective than dropout in this case and can perform better than STOA listwise approach SQL-Rank [32] on 2 datasets out of 3.

In Table 2, SSE-SE has two tuning parameters: probability p_u to replace embeddings associated with user-side embeddings and probability p_i to replace embeddings associated with item side embeddings because there are two embedding tables. But here for simplicity, we use one tuning parameter $p_s = p_u = p_i$. We use dropout probability of p_d, dimension of user/item embeddings d, weight decay of λ and learning rate of 0.01 for all experiments, with the exception that the learning rate is reduced to 0.005 when both SSE-SE and Dropout are applied. For Douban dataset, we use $d = 10, \lambda = 0.08$. For Movielens10m and Netflix dataset, we use $d = 50, \lambda = 0.1$.

5.2 Neural Machine Translation

We use the transformer model [30] as the backbone for our experiments. The control group is the standard transformer encoder-decoder architecture with self-attention. In the experiment group, we apply SSE-SE towards both encoder and decoder by replacing corresponding vocabularies' embeddings in the source and target sentences. We trained on the standard WMT 2014 English to German dataset which consists of roughly 4.5 million parallel sentence pairs and tested on WMT 2008 to 2018 news-test sets. Sentences were encoded into 32,000 tokens using a byte-pair encoding. We use the SentencePiece, OpenNMT and SacreBLEU implementations in our experiments. We trained the 6-layer transformer base model on a single machine with 4 NVIDIA V100 GPUs for 20,000 steps. We use the same dropout rate of 0.1 and label smoothing value of 0.1 for the baseline model and our SSE-enhanced model. Both models have dimensionality of embeddings as $d = 512$. When decoding, we use beam search with the beam size of 4 and length penalty of 0.6 and replace unknown words using attention. For both models, we average last 5 checkpoints (we save checkpoints every 10,000 steps) and evaluate the model's performances on the test datasets using BLEU scores. The only difference between the two models is whether or not we use our proposed SSE-SE with $p = 0.01$ in Equation 6 for both encoder and decoder embedding layers.

5.3 BERT

In the first leg of experiments, we crawled one million user reviews data from IMDB and pre-trained the BERT-Base model (12 blocks) for $500,000$ steps using sequences of maximum length 512 and batch size of 8, learning rates of $2e^{-5}$ for both models using one NVIDIA V100 GPU. Then we pre-trained on a mixture of our crawled reviews and reviews in IMDB sentiment classification tasks (250K reviews in train and 250K reviews in test) for another $200,000$ steps before training for another $100,000$ steps for the reviews in IMDB sentiment classification task only. In total, both models are pre-trained on the same datasets for $800,000$ steps with the only difference being our model utilizes SSE-SE. In the second leg of experiments, we fine-tuned the two models obtained in the first-leg experiments on two sentiment classification tasks: IMDB sentiment classification task and SST-2 sentiment classification task. The goal of pre-training on IMDB dataset but fine-tuning for SST-2 task is to explore whether SSE-SE can play a role in transfer learning.

The results are summarized in Table 6 for IMDB sentiment task. In experiments, we use maximum sequence length of 512, learning rate of $2e^{-5}$, dropout probability of 0.1 and we run fine-tuning for 1 epoch for the two pre-trained models we obtained before. For the Google pre-trained BERT-base model, we find that we need to run a minimum of 2 epochs. This shows that pre-training can speed up the fine-tuning. We find that Google pre-trained model performs worst in accuracy because it was only pre-trained on Wikipedia and books corpus while ours have seen many additional user reviews. We also find that SSE-SE pre-trained model can achieve accuracy of 0.9542 after fine-tuning for one epoch only. On the contrast, the accuracy is only 0.9518 without SSE-SE for embeddings associated with output y_i.

For the SST-2 task, we use maximum sequence length of 128, learning rate of $2e^{-5}$, dropout probability of 0.1 and we run fine-tuning for 3 epochs for all 3 models in Table 7. We report AUC, accuracy and F1 score for dev data. For test results, we submitted our predictions to Glue website for the official evaluation. We find that

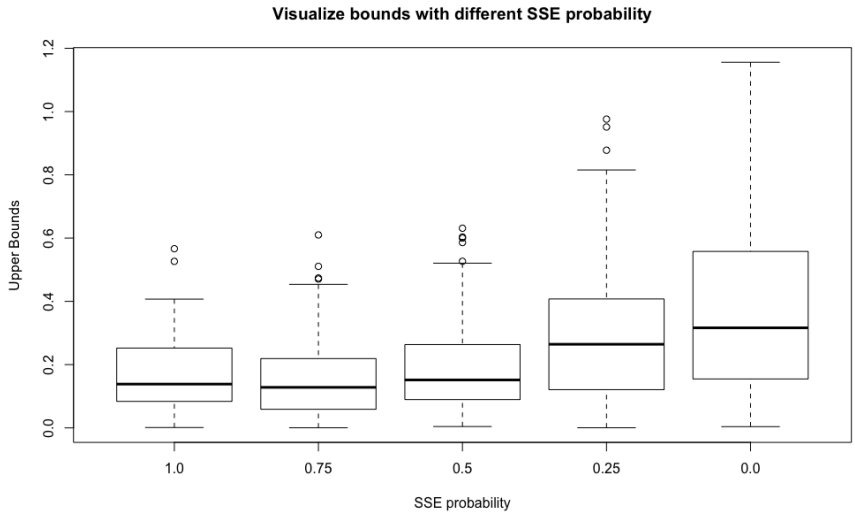

Figure 5: Simulation of a bound on $\rho_{L,n}$ for the movielens1M dataset. Throughout the simulation, L is replaced with ℓ (which will bound $\rho_{L,n}$ by Jensen's inequality). The SSE probability parameter dictates the probability of transitioning. When this is 0 (box plot on the right), the distribution is that of the samples from the standard Rademacher complexity (without the sup and expectation). As we increase the transition probability, the values for $\rho_{L,n}$ get smaller.

even in transfer learning, our SSE-SE pre-trained model still enjoys advantages over Google pre-trained model and our pre-trained model without SSE-SE. Our SSE-SE pre-trained model achieves 94.3% accuracy on SST-2 test set versus 93.6 and 93.8 respectively. If we are using SSE-SE for both pre-training and fine-tuning, we can achieve 94.5% accuracy on the SST-2 test set, which approaches the 94.9 score reported by the BERT-Large model. SSE probability of 0.01 is used for fine-tuning.

5.4 Proofs

Throughout this section, we will suppress the probability parameters, $p(.,.|\Phi) = p(.,.)$.

Proof of Theorem 1. Consider the following variability term,

$$\sup_{\Theta} |S(\Theta) - S_n(\Theta)|. \tag{14}$$

Let us break the variability term into two components

$$\mathbb{E}_{X,Y} \sup_{\Theta} \left|S_n(\Theta) - \mathbb{E}_{Y|X}[S_n(\Theta)]\right| + \mathbb{E}_{X,Y} \sup_{\Theta} \left|\mathbb{E}_{Y|X}[S_n(\Theta)] - S(\Theta)\right|,$$

where X, Y represent the random input and label. To control the first term, we introduce a ghost dataset (x_i, y_i'), where y_i' are independently and identically distributed according to $y_i|x_i$. Define

$$S_n'(\Theta) = \sum_i \sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) \ell(\mathbf{E}[\mathbf{k}], y_i'|\Theta) \tag{15}$$

be the empirical SSE risk with respect to this ghost dataset.

We will rewrite $\mathbb{E}_{Y|X}[S_n(\Theta)]$ in terms of the ghost dataset and apply Jensen's inequality and law of iterated conditional expectation:

$$\mathbb{E} \sup_{\Theta} \left|\mathbb{E}_{Y|X}[S_n(\Theta)] - S_n(\Theta)\right| \tag{16}$$

$$= \mathbb{E} \sup_{\Theta} \left|\mathbb{E}_{Y'|X}\left[S_n'(\Theta) - S_n(\Theta)\right]\right| \tag{17}$$

$$\leq \mathbb{E}\mathbb{E}_{Y'|X}\left[\sup_{\Theta} \left|S_n'(\Theta) - S_n(\Theta)\right|\right] \tag{18}$$

$$= \mathbb{E} \sup_{\Theta} \left|S_n'(\Theta) - S_n(\Theta)\right|. \tag{19}$$

Notice that

$$S'_n(\Theta) - S_n(\Theta) = \sum_i \sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) \left(\ell(\mathbf{E}[\mathbf{k}], y'_i|\Theta) - \ell(\mathbf{E}[\mathbf{k}], y_i|\Theta)\right)$$
$$= \sum_i \sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) \left(e(\mathbf{E}[\mathbf{k}], y'_i|\Theta) - e(\mathbf{E}[\mathbf{k}], y_i|\Theta)\right).$$

Because $y_i, y'_i|X$ are independent the term $(\sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) \left(e(\mathbf{E}[\mathbf{k}], y'_i|\Theta) - e(\mathbf{E}[\mathbf{k}], y_i|\Theta)\right))_i$ is a vector of symmetric independent random variables. Thus its distribution is not effected by multiplication by arbitrary Rademacher vectors $\sigma_i \in \{-1, +1\}$.

$$\mathbb{E} \sup_{\Theta} \left|S'_n(\Theta) - S_n(\Theta)\right| = \mathbb{E} \sup_{\Theta} \left| \sum_i \sigma_i \sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) \left(e(\mathbf{E}[\mathbf{k}], y'_i|\Theta) - e(\mathbf{E}[\mathbf{k}], y_i|\Theta)\right) \right|.$$

But this is bounded by

$$2\mathbb{E}\mathbb{E}_\sigma \sup_{\Theta} \left| \sum_i \sigma_i \sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) e(\mathbf{E}[\mathbf{k}], y_i|\Theta) \right|.$$

For the second term,

$$\mathbb{E} \sup_{\Theta} \left|\mathbb{E}_{Y|X}[S_n(\Theta)] - S(\Theta)\right|$$

we will introduce a second ghost dataset x'_i, y'_i drawn iid to x_i, y_i. Because we are augmenting the input then this results in a new ghost encoding $\mathbf{j}'^i$. Let

$$S'_n(\Theta) = \sum_i \sum_{\mathbf{k}} p(\mathbf{j}'^i, \mathbf{k}) \ell(\mathbf{E}[\mathbf{k}], y'_i|\Theta) \tag{20}$$

be the empirical risk with respect to this ghost dataset. Then we have that

$$S(\Theta) = \mathbb{E}_{X'} \mathbb{E}_{Y'|X'} S'_n(\Theta)$$

Thus,

$$\mathbb{E} \sup_{\Theta} \left|\mathbb{E}_{Y|X}[S_n(\Theta)] - S(\Theta)\right| \tag{21}$$
$$= \mathbb{E} \sup_{\Theta} \left|\mathbb{E}_{X'} \left[E_{Y|X}[S_n(\Theta)] - E_{Y'|X'}[S'_n(\Theta)]\right]\right| \tag{22}$$
$$\leq \mathbb{E}\mathbb{E}_{X'} \left[\sup_{\Theta} \left|E_{Y|X}[S_n(\Theta)] - E_{Y'|X'}[S'_n(\Theta)]\right|\right] \tag{23}$$
$$= \mathbb{E} \sup_{\Theta} \left|E_{Y|X}[S_n(\Theta)] - E_{Y'|X'}[S'_n(\Theta)]\right|. \tag{24}$$

Notice that we may write,

$$E_{Y|X}[S_n(\Theta)] - E_{Y'|X'}[S'_n(\Theta)] = \sum_i \sum_{\mathbf{k}} \left(p(\mathbf{j}^i, \mathbf{k}) - p(\mathbf{j}'^i, \mathbf{k})\right) L(\mathbf{E}[\mathbf{k}]|\Theta)$$

Again we may introduce a second set of Rademacher random variables σ'_i, which results in

$$\mathbb{E} \sup_{\Theta} \left|E_{Y|X}[S_n(\Theta)] - E_{Y'|X'}[S'_n(\Theta)]\right| \leq 2\mathbb{E}\mathbb{E}_{\sigma'} \sup_{\Theta} \left| \sum_i \sigma'_i \sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) L(\mathbf{E}[\mathbf{k}]|\Theta) \right|.$$

And this is bounded by

$$2\mathbb{E}\mathbb{E}_{\sigma'} \sup_{\Theta} \left| \sum_i \sigma'_i \sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) L(\mathbf{E}[\mathbf{k}]|\Theta) \right| \leq 2\mathbb{E} \sup_{\Theta} \left| \sum_i \sigma'_i \sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) \ell(\mathbf{E}[\mathbf{k}], y_i|\Theta) \right|$$

by Jensen's inequality again. □

Proof of Theorem 2. It is clear that $2\mathcal{B} \geq B(\hat{\Theta}) + B(\Theta^*)$. It remains to show our concentration inequality. Consider changing a single sample, (x_i, y_i) to (x'_i, y'_i), thus resulting in the SSE empirical risk, $S_{n,i}(\Theta)$. Thus,

$$S_n(\Theta) - S_{n,i}(\Theta) = \sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) \cdot \ell(\mathbf{E}[\mathbf{k}], y_i|\Theta) - \sum_{\mathbf{k}} p(\mathbf{j}'^i, \mathbf{k}) \cdot \ell(\mathbf{E}[\mathbf{k}], y'_i|\Theta)$$
$$= \sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) \cdot \left(\ell(\mathbf{E}[\mathbf{k}], y_i|\Theta) - \ell(\mathbf{E}[\mathbf{k}], y'_i|\Theta)\right) + \sum_{\mathbf{k}} \left(p(\mathbf{j}'^i, \mathbf{k}) - p(\mathbf{j}^i, \mathbf{k})\right) \cdot \ell(\mathbf{E}[\mathbf{k}], y'_i|\Theta)$$
$$\leq b \left(\sum_{\mathbf{k}} p(\mathbf{j}^i, \mathbf{k}) + \sum_{\mathbf{k}} p(\mathbf{j}'^i, \mathbf{k}) \right) \leq 2b.$$

Then the result follows from McDiarmid's inequality.

□