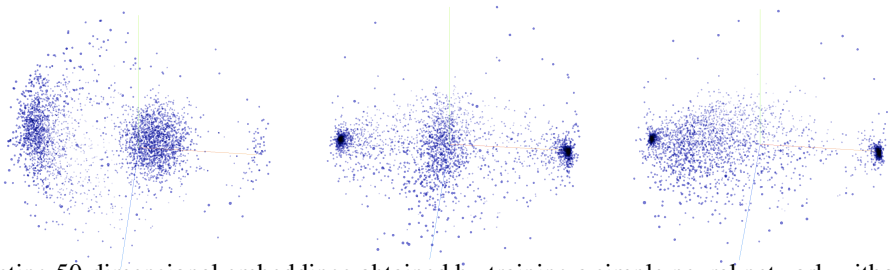

Figure 1: Projecting 50-dimensional embeddings obtained by training a simple neural network without SSE (Left), and with SSE-Graph (Center) , SSE-SE (Right) into 3D space using PCA.

Table 1: The experimental results of BERT-NER with SSE-SE and without SSE-SE when we only allow fine-tuning BERT for one epoch and two epochs. The same set of hyper-parameters are used.

	One Epoch				Two Epochs			
Model	accuracy(%)	precision(%)	recall(%)	F1 Score	accuracy(%)	precision(%)	recall(%)	F1 Score
BERT-NER	97.85	86.87	88.30	87.58	98.02	88.13	90.10	89.11
BERT-NER + SSE-SE	**97.91**	**87.80**	**88.57**	**88.18**	**98.07**	**88.16**	**90.50**	**89.31**

We thank the reviewers for their insightful feedback. In the following, we address their concerns and questions.

R1. It is indeed a great suggestion to examine concrete examples beyond the quantitative evaluation to get an intuition. In Figure 1 (also in Figure 3 in the appendix), we use an embedding projection tool to visualize the embeddings from Matrix Factorization model after applying dimension reduction to 3D space. When looking at some specific examples, e.g. movies starred by the same actors, corresponding embeddings are closer to each other after using SSE-Graph than the version without SSE. That is likely due to the use of item graph. Interestingly, even for SSE-SE, it does make movies of the same genre closer to each other since it implicitly allows information sharing between embeddings.

R2. Thanks Reviewer 2 for raising the valuable question about the interpretation of SSE and connection to dropout. So far, we have only explored the bias and variance trade-off. As shown in Theorem 1, SSE can 'smooth' the Rademacher complexity. We have not studied whether SSE can be interpreted as Gaussian noise injection, but we feel it might have similar effects. Figure 1 shows that the distributions of embeddings are definitely affected by the SSE-Graph and SSE-SE and perhaps we can further study how this is related to dropout in theory.

R3. Following your suggestions and concerns, we used the BERT-NER with CRF loss to conduct the experiments in Table 1. We did not have time to do pre-training, so this is only applying SSE-SE towards the fine-tuning stage of NER. We fix the same set of hyper-parameters and use the same CoNLL-2003 training/test split. SSE probability of 0.01 is used. Table 1 seems to imply that even for a task like NER, SSE-SE can still benefit the training procedure. Moreover, for NER, it is perhaps intuitively reasonable to construct a graph over the vocabulary in the absence of Knowledge Graphs, by connecting tokens of the same type: for example, an edge exists between two tokens if they are both Persons.

We agree that we need to dive deeper into each specific task and understand better when SSE helps. Our contention is that SSE can help when the embedding space is very large, which does occur often in recommendation systems, but also in NLP. It is outside of the scope of this paper to prove the usefulness of SSE to very different domains such as computer vision. Currently, on all the tasks we tried (with large embedding space), SSE can always at least slightly improve the performance; while for some tasks SSE can lead to significant performance gains. In general, we find SSE-Graph and SSE-SE helps most when the model gets over-parameterized, either there is a large number of embeddings or the dimensionality of embeddings is very large while the number of training data is limited. Moreover, we find that SSE-Graph and SSE-SE can generally speed-up the training procedure of the original methods without the need to touch the learning rate (as shown in Figure 2 in original submission) and that the existence of an informative knowledge graph can often make this speed-up more significant.

We have also compared SSE-Graph with standard Graph Laplacian Regularization in Table 2 and we find that using SSE-Graph can achieve much better RMSE. Moreover, it would not create a significant overhead to the training procedure of Matrix Factorization as Graph Laplacian Regularization did.

Table 2: Compare SSE-Graph against Graph Laplacian Regularization.

	Movielens1m	Movielens10m
Model	RMSE	RMSE
SGD-MF	1.0984	1.9490
Graph Laplacian + ALS-MF	1.0464	1.9755
SSE-Graph + SGD-MF	**1.0145**	**1.9019**