**Reviewer 1**: Thank you for the insightful analysis and acknowledgement of our effort.

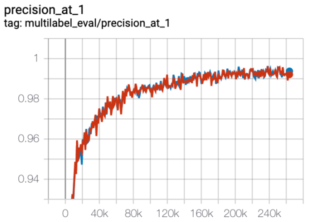

Figure 1: Training (blue) and test (red) accuracy on the EU-RLEX dataset when trained on the union of training and test data.

**Reviewer 2**: **Re Quality**: Empirically we observe overfitting during training on all public datasets, which Reviewer 3 has also mentioned. We chose Amazon-13K as a representative dataset to demonstrate this phenomenon, but can certainly include plots on other datasets in the supplementary material for further evidence.

**Re Soundness**: Given enough capacity, a two-layer neural network (such as our model in Sec. 4) is theoretically guaranteed to be able to represent *any* continuous mapping since it is a universal approximator (Hornik, 1991). The question here is whether the embedding model will overfit to limited samples, which is our main message and is addressed by the GLaS regularizer.

**Re Clarity and organization:** Due to limited space, we could only include the most relevant prior works but can certainly include additional details in the supplementary. We moved less relevant details to the supplementary (including the pseudo-code) to include more experimental results. We will split the table to improve readability and reorganize the pseudo-code in the main text in the final version using the extra granted page if the paper gets accepted.

**Re Training on the union of training and test data**: Fig. 1 shows P@k across training epochs when we train on the union of training and test data. The model is clearly expressive enough as training and test accuracy are near-perfect.

**Reviewer 3**: **1.** XMC datasets have been well-researched and improvements ***“by couple of % points"*** are significant. For example, Parabel (WWW ’18) only improved P@1 on WikiLSHTC(+0.64%) over the previous best but we significantly improved P@k SOTA on 3 datasets and PSP@k SOTA on 4 datasets. Regarding the use of deep learning, note that our method outperforms XML-CNN in Table 3 that uses both convolutional and fully-connected layers. Hence, our work is a matter of using an appropriate (even simple) architecture, loss function, optimization, and regularization. Our main goal is to debunk the low-dimensional bottleneck misconception by demonstrating this through a simple neural network model with a novel regularization framework.

**2.** It is true that LEML and SLEEC use similar architectures, but there are dramatic differences in the choice of the loss function: LEML uses a least square regression loss, whereas SLEEC uses a nearest neighbor loss. In contrast, our approach uses a margin-based loss complemented by stochastic optimization and novel regularization. While we agree that it is certainly informative to analyze where the crucial difference lies, we believe it is also important to highlight the main message of the paper, namely that proper design and training of embedding-based models can enable them to outperform other approaches.

**3.** We did not intend to make this impression that over-fitting is new or surprising for these datasets. Our goal was to show that we should not attribute the poor performance of embedding-based methods to the low-dimensional bottleneck, in direct response of the following quote from the DiSMEC paper, *"In XMC setting which consists of a diverse power-law distributed label space, the crucial assumption made by the embedding-based approaches of a low rank label space breaks down."* In Sec 2.2 and Theorem 2.1, we rigorously showed the existence of a perfect accuracy low-dimensional embedding-based classifier and the possibility of over-fitting with small training sets.

**4.** It is not necessary for PSP@k metrics to correlate very well with P@k and our results are not the first to be *“surprising.”* **For example, compare P@k and PSP@k of PfastreXML and FastXML in Table 3 and Table 4. We also respectfully disagree with the reviewer’s quote *"the proposed method does nothing special for tail-labels."*** In lines 204-206 we have mentioned that because of tail labels we regularize the label embeddings to be near-orthogonal by Eq. (2). Note that near-orthogonality is condition No. 5 mentioned in Theorem 2.1 for the existence of a perfect embedding-based classifier. GLaS regularization corrects over-penalizing based on the co-occurrence of labels which is indeed correlated with algebraic connectivity. As label co-occurrence or algebraic connectivity gets smaller (consider Amazon670k, WikiLSHTC, EURLex in Table 2 of arXiv:1803.01570), we get better PSP@k improvement over P@k because of having more near-orthogonal embedding and less GLaS correction. Due to the lack of space, we were not able to include our code here. However, our code is a TensorFlow translation of the MATLAB code provided in the XMC repository and we have verified its correctness and we will release it in the final version.

**5.** The values of $d$ and $t$ are chosen so that the sum of the probabilities in lines 474 and 475 is less than 1, which implies that with probability $> 0$, neither of the two events happens, from which the statement of line 478 follows.

**6.** We perform training on a cluster of servers with 2 Intel Xeon CPUs and inference with a single thread on a single server. We accelerate inference with approximate inner product search algorithms to bring the inference time to below 10 ms for large datasets. We are happy to provide more details if this is a point of concern.

**7.** We mainly relied on the XMC repository for the baselines but will for sure cite these references. Re ProXML, note that our PSP@k results outperform the ones in the ProXML paper. Re AttentionXML, it is a tree-based model that unlike all other baselines use raw text features and our method outperforms it on PSP@k metric (Fig 2 of AttentionXML paper on Amazon670k dataset). Also, please note that according to Hugo Larochelle (NeurIPS PC) *"it is not reasonable to compare current NIPS submissions with work that hasn’t been accepted at a venue prior to submission."*