We thank all the reviewers for their time and insightful feedback about our work. We address one of the core challenges in training machine learning models with limited labels. This is crucially important for tasks with sensitive user data where we cannot manually access and annotate a lot of data, as well as for low-resource tasks in different languages. Many of the recent few-shot learning works focus on computer vision compared to NLU tasks. Despite the promise of pre-trained language models in overcoming the annotation bottleneck, we still see a gap in performance when these models are trained on a few samples (say, 20-30 samples in our setting) in contrast to thousands of annotations with an average accuracy gap of **14%** for tasks in our work. We leverage self-training with several advances to bridge this gap.

R1 (Q1) raises an important point with respect to developing a sound annotation scheme. Note that even if we resolve this challenge, it is still too expensive and in some cases infeasible to obtain large-scale human annotations for many specialized domains especially dealing with sensitive data. **(Q2)** From Table 2, we observe our method to be **4.2%** better than classic self-training and **2.2%** better than UDA. Note that UDA has access to a Neural Machine Translation system that generates paraphrases for consistency learning, whereas our model does not leverage any such external resource, and, therefore, is more general. Table 4 compares different models with the same setup as ours leveraging different forms of pre-training. We observe our model to obtain atleast **7%** improvement in IMDB and **4%** improvement in AG News over our closest baseline in the form of variational pre-training [Gururangan et al., 2019] and reinforcement learning with adverarial training [Li and Ye, 2018], while using **3x-6x less** training labels (shown by **K** in **Table 4**). **(Q3)** Table 2 reports the accuracy numbers averaged over runs with *multiple* random seeds for fair comparison across different models. While in Figure 2, we *fix* the random seed (for demonstration) and change other parameters within our model to show variations with different number of training labels and self-training accuracy over iterations.

R2 (Q1) The reviewer raises a good point regarding a simpler selection strategy that can be used as baseline with classic ST. Similar baselines reported for active learning [Gal et al., 2017] and preference learning [Houlsby et al., 2011] show the BALD measure outperforming them. Noting the above concern, we experimented with classic ST with confidence-based and class-dependent sample selection (as suggested by the reviewer) where confidence is given by predicted class probabilities. Preliminary experiments (over several runs with different seeds) show classic ST with such selection strategy to perform marginally better (0.5% acc. improvement with some task-specific variance) than classic ST (without selection) on an average in the few-shot setting (we will report detailed numbers in paper). Classic ST performs unbiased sample selection with uniform sampling forming a competitive baseline (often ignored in many works). Confidence-based sample selection (ignoring uncertainty) relies on the most confident predictions from a *weak* teacher resulting in early drifts from noisy pseudo-labels [Zhang et al., 2017]. This results from our few-shot setting where the teacher is fine-tuned on few labeled samples to start with, in contrast to many works employing such strategies with a stronger teacher. **(Q2)** Confident learning (Equation 11) incorporates sample variance (modeled by **Var(y)**) with minimization objective for *explicit* reduction. *Implicit* variance reduction happens via selecting samples with low uncertainty for self-training. Classic ST cannot use sample variance without using model uncertainty (that we achieve using MC dropout). Earlier baseline in *(Q1)* can be used to derive sample mean, but not the variance without accessing historical behavior or information from several stochastic passes. **(Q3)** raises an important concern regarding UDA. Consider the following differences. (1) Publicly available UDA code does not use validation set, instead, reports the maximum (across all epochs) and the last epoch accuracy on test set. We report UDA results on test set from the model with the best validation accuracy. (2) Recent works on data augmentation like SimCLR [Chen et al., 2020], UDA [Xie et al., 2019] and self-training with noisy student [Xie et al., 2020] show these techniques to work best with large batch sizes *as also applicable to our model*. Additionally, for IMDB longer sequence length plays a big role. For a fair comparison, with access to same amount of computational resources, we report UDA results and ours on the same hardware (V100 GPU) with maximum permissible batch-size and sequence length for every model. Due to page limitations, these settings were discussed in Appendix (lines 15-21) and will be moved to the main table as per suggestion. **(Q4)** As per our description in lines 186-189, Equation 12 should read $log(Var(y))$ (we will fix this typo).

R3 Thanks for the feedback and suggestions. We are extending this work for more real-world tasks including multilingual settings where large-scale human annotations are difficult to obtain.

R4 (Q1) raises an important point regarding our simulation of the few-shot setting with in-domain unlabeled data (as also used in prior work). Extending these models to more realistic low-resource tasks with proxy/noisy data from related domains is an exciting direction for future work. **(Q2)** As per our description in lines 186-189, Equation 12 should read as $log(Var(y))$. Negative signs for cross-entropy loss and log inverse cancel out. **(Q3)** We do not need to estimate σ for the minimization objective since it is independent of y. **(Q4)** Hard pseudo-labels are optimized with cross-entropy loss similar to hard ground-truth labels. **(Q5)** Sample mixing based on *easy* and *hard* examples is an interesting idea. We explored something similar with mixing equal number of instances sampled with BALD measure and remaining with uniform sampling. This presented mixed results that performed marginally better than ours for some of the runs with different seeds – warranting further exploration. **(Q6)** For tasks like DBpedia and Elec with very high performance given few training labels, there is diminishing returns on injecting more labels. In contrast, we improve more for tasks that are comparatively difficult like IMDB (very long reviews), AG News (4-class) and SST (very short snippets).