Authors' response for paper "When Does Label Smoothing Help?"

R1, R2, R3: Thanks for taking the time to review our paper and for providing thoughtful feedback.

R2: *"In Figure 4, why does changing the temperature of a network...with label smoothing degrade calibration?"*: Temperature scaling is applied after training to correct the calibration of the predictions by dividing the logits by a scalar (temperature), but label smoothing yields calibrated predictions without this additional step. Temperature scaling of an already calibrated network will only degrade ECE.

R2: *"In Figure 5, why does label smoothing slightly degrade the baseline performance of the student network?"*: It is hard to separate the effect of one hyperparameter from the many others that affect performance (learning rate schedule, weight decay, etc.), as they are not completely independent. In this experiment, the goal was to compare the relative performance of training the student from label smoothing targets compared to targets provided by a teacher. For fair relative performance comparison, both of these cases have the same training schedule, hyperparameters and equivalent probability mass distributed among incorrect classes. Although in this particular experiment label smoothing hurts performance, in the best performing models in a variety of tasks (Table 1), label smoothing gives consistent gains.

R2: *"label smoothing could be particularly useful for generalization on samples from classes that are semantically similar"*: We investigated (same setup as submission) the confusion matrix for the CIFAR-100 dataset, which comprises 20 super-classes containing 5 classes each. Over 5 different runs, we compare the errors made by a network trained with hard targets (2786 ± 29) which can be divided into "fine" errors in the same super-class (1209 ± 13, 43.4% of total errors) and "coarse" errors in a different super-class (1577 ± 30). With label smoothing, the total # of errors is reduced (2732 ± 34) with a reduction of "coarse" errors (1515 ± 25) while the "fine" errors remains close to constant (1217 ± 17) representing 44.6% of the total errors. While reduction of only coarse errors is not intuitive, we observe that the ratio of "fine" errors remains stable and we believe more experiments should be done to verify if the effect is consistent.

R2: *"I would suggest reorganizing the layout of the paper"*: We agree that the effect of label smoothing on calibration and distillation has direct practical applications, but we also feel the visualization provides useful intuition. That said, we are open to revisiting the ordering in the camera-ready version.

R3: *"Conduct more empirical analysis of other tasks to verify the findings on label smoothing"*: Below, we show visualizations for the English-to-German translation task. The results are similar to the image classification visualization results in the submission. Next-token prediction is equivalent to classification: In image classification we maximize the likelihood $p(y|x)$ of the correct class given an image, whereas in translation, we maximize the likelihood $p(y_t|x, y_{0:t-1})$ of next token given source sentence and preceding target sequence. However, there are differences between the tasks that affect visualization and distillation.

- The image classification datasets we examine have balanced class distributions, whereas token distributions in translation are highly imbalanced. (This could potentially be addressed with unigram label smoothing.)
- For image classification, we can get near-perfect training set accuracy and still generalize; in this case, label smoothing erases information, whereas hard-targets preserves it. In the translation task, next-token accuracy on the training set is around 80%. Therefore, visualizations show errors in both the training and validation sets (tending to tight clusters as expected). For distillation, this means a student may learn from the teacher's errors. Since teachers trained with and without label smoothing will both have errors that the student can learn from, it is unclear which will perform best.
- For image classification, the penultimate layer dimension is usually higher than the number of classes, so templates can lie on a regular simplex (i.e. equidistant to each other). In translation, we have a ~30k token alphabet in 512 dimensions, so a simplex is not possible.

In our submission, for visualization and distillation, we concentrate on the image classification case to simplify intuition and experimental design. The visualization results below demonstrate that some features of the image classification visualization also hold for translation. We leave analysis of distillation for translation for future work. We will include a section dedicated to the differences between these cases in the appendix, as well as the figure we provide here.

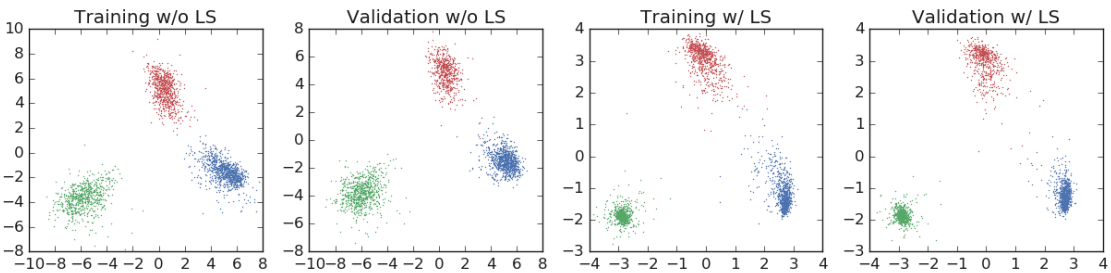

Figure 1: Visualizations of penultimate representations of Transformer trained to perform English-to-German translation.