ACCURATE LAYERWISE INTERPRETABLE COMPETENCE ESTIMATION (ALICE)

VICKRAM RAJENDRAN AND WILLIAM LEVINE

PROBLEMS IN DEPLOYING MACHINE LEARNING

- We need ground truth in order to test our machine learning models…
  - Normally we test with held aside test sets, but these must be labeled to calculate metrics.
  - Often we can't trust our metrics to generalize to the real world due to distributional shift – but we don't have ground truth after deployment to evaluate our models there.
- Problem: Test set metrics don't tell us how well we'll do on a particular point in the real world.
- Problem: Real world data points are not labeled, so we can't tell if our model is performing well or not.
- Problem: A machine learning model "doing well" is dependent on the particular use-case.
  - Some models can have large margins of error and still be performing competently, while others must be very accurate.

PROBLEMS IN DEPLOYING MACHINE LEARNING

- We need a way to tell when a model is performing competently, without having access to ground truth.
  - This should encompass "all" definitions of competence, regardless of use-case.

- This is essentially a generalized form of **uncertainty estimation:**
  - We want to estimate the probability that our model is competent or incompetent on a particular point, or the uncertainty in our model.
- *For now, we'll restrict ourselves to classification models.*

COMPETENCE: GENERALIZED CONFIDENCE

- Given a model $\hat{f}$ that approximates the true classification model f, **confidence** is normally defined as $p\left(\hat{f}(x) = f(x) \mid \hat{f}, x\right)$, or the probability that your model is correct.
- We'll generalize this to not just being correct, but to have your "error function" E be less than some threshold δ. So in this case, confidence is the same as

$$p\left(E_{0,1}\left(\hat{f}(x), f(x)\right) < \delta\right)$$

where $E_{0,1}$ is the 0-1 error function (0 when $\hat{f}$ and f are the same on x, 1 otherwise), and δ is anything between 0 and 1.

COMPETENCE: GENERALIZED CONFIDENCE

- Competence will be allowing the error functions and thresholds to change. Given an error function E and a threshold δ, We define the **competence** of a model $\hat{f}$ on a point x to be $p\left(E\left(\hat{f}(x), f(x)\right) < \delta | \hat{f}, x\right)$.
- Given an ϵ, a model is $\boldsymbol{\delta - \epsilon}$ **competent** if this probability is greater than ϵ.
- This notion is the same as the model being Probably (competence greater than a threshold) Approximately Correct (error is less than delta).

RELATED WORK: UNCERTAINTY ESTIMATION

- Okay, we've defined competence to allow for lots of different use-cases by letting the user choose their error function and thresholds…
- How do we estimate it? Most people do uncertainty estimation with:
- 1) Bayesian Neural Networks – Requires specific architecture, harder to scale to large problems, and doesn't allow for changing our definition of competence on the fly (requires recomputation of posteriors)
- 2) Ensembling Methods – Requires specific architectures, harder to scale to large problems.

RELATED WORK: UNCERTAINTY ESTIMATION

- Most methods don't generalize to all classification architectures (i.e. they require a particular architecture, such as having Dropout (MC Dropout) or BNN's or ensembles), or they use Out of Distribution Data (Like Prior Networks).
- All of these methods also just estimate *confidence*, rather than general *competence*.
- Further, most of these methods perform poorly in areas of large model uncertainty (model fits poorly to data), data uncertainty (class overlap/label noise), or distributional uncertainty (facing inputs it has never seen before).
- We want a robust method to estimate competence that generalizes to everything – regular ML models and deep models regardless of architecture, and takes into account all aspects of predictive uncertainty. We'd also like it to be fast ☺

ALICE

- We derived a new competence estimator for any classification model, regardless of error functions, delta, and architecture. See our paper for the details of the derivation, but we essentially approximate distributional, data, and model uncertainty in turn.

ALICE ACRONYM

- Accurate:
  - We'll see in the coming slides that ALICE is an accurate competence estimator (it predicts pretty well whether or not a model is competent or incompetent on a new test point)
- Layerwise:
  - ALICE can be computed on individual layers. Future work will include aggregating the ALICE scores of individual layers together.
- Interpretable:
  - ALICE is an honest probability – it's calibrated, which means the ALICE score, unlike the Trust Score or Softmax, is actually interpretable.
- Competence
  - ALICE works on all aspects of competence, not just confidence like other uncertainty estimators.
- Estimation
  - ALICE doesn't need ground truth in order to Estimate the competence of a machine learning model.

ALICE: APPROXIMATING DISTRIBUTIONAL UNCERTAINTY OF A POINT

- Similar to related work, we fit class-conditional Gaussian's to each class' input points, and then compute the mahalanobis distance between our target point x and these Gaussians. To turn this into a probability, we take the empirical distribution of the mahalanobis distances of a validation set and compute p(D|x), the probability that x is in distribution, as {max 1 – CDF(mahalanobis distance(x)} over all of the class-conditional Gaussian's.

ALICE: APPROXIMATING DATA AND MODEL UNCERTAINTY

- We use an indicator function to determine whether or not the output of the model c*ould* be competent if the true class was a particular class j , and then use a calibrated logistic regressor as a transfer classifier to determine the probability that the true class is class j, given that the point is in distribution.

ALICE: PUTTING IT ALL TOGETHER

- The ALICE score is a lower bound for the competence of a model. Putting the previous slides together, we get:

$$p(E < \delta) \approx p(D) \sum_{j} \quad I\left(E\left(\hat{f}, c_j\right) < \delta\right) p\left(c_J \middle| D\right)$$

Where p(D) is the distributional uncertainty term and p(c_j| D) is the logistic regressor. See our paper for more details.

NOTE: ALICE is fast to calculate! The initial fitting can be slow, but afterwards it's just a single LR inference plus vectorized mahalanobis calculations.

NOTE: ALICE can happen at each individual layer of a neural network! We would just compute the Gaussian's

EVALUATING COMPETENCE ESTIMATORS

- A competence estimator can predict whether a model is competent on a point, and with ground truth we can compute whether a model is truly competent.
- This is just a binary classification problem! We can just use Binary Classification metrics to evaluate competence estimators.
- Given an error function, we'll compute the average precision at a particular delta, and then take the mean of these AP's across a range of 100 deltas.
- This metric essentially tests how well the estimator can **rank** the points, from least competent to most competent.

EVALUATING COMPETENCE ESTIMATORS

- To test calibration, we use calibration curves. We bin the scores of the competence estimator into ten equally spaced bins, and then compute what proportion of points in each bin were competent.

UNCERTAINTY ESTIMATION COMPARISONS

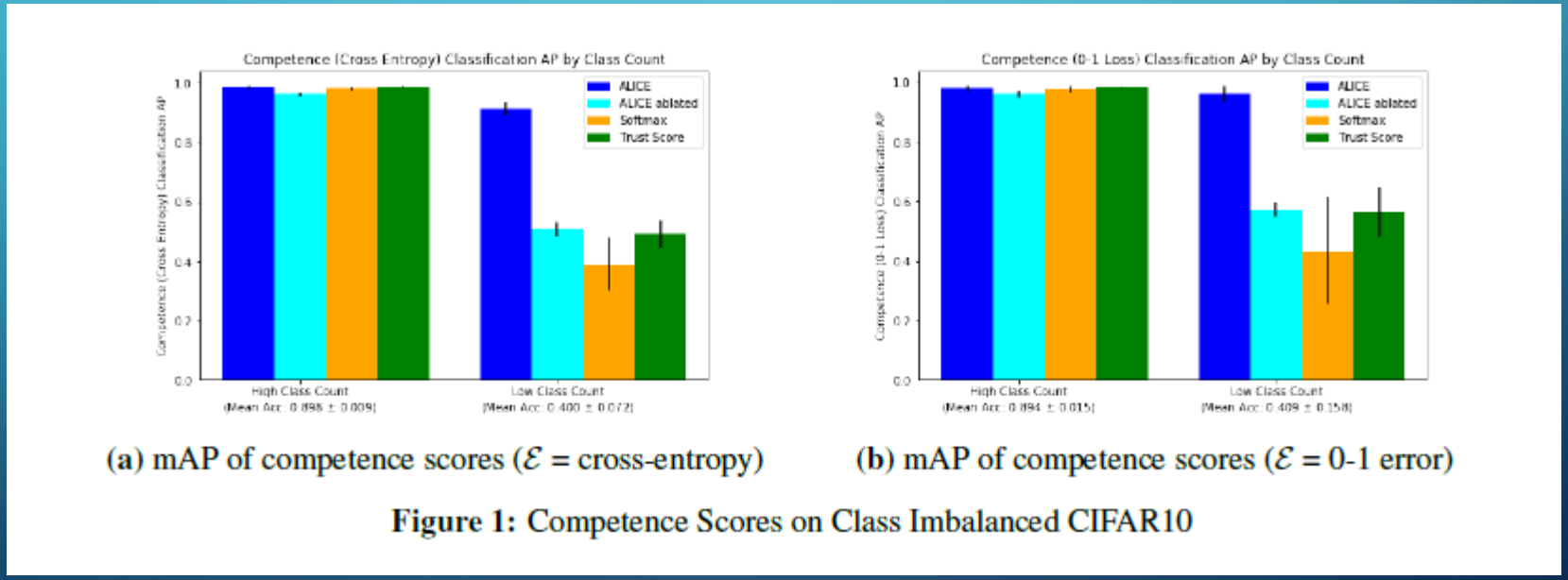
Of the ones that does work on every classification model, we'll use:

- Softmax – The baseline notion of model confidence.
- Trust Score – A NeurIPS 2018 paper that outperforms softmax pretty much across the board in predicting whether or not a classification model will be correct or incorrect.

- These work on all architectures and don't require any out of distribution data.
- NOTE: Neither of these methods are *calibrated*, so we don't compute calibration curves on them.

MODEL UNCERTAINTY

- ALICE can predict competence in areas of model uncertainty:



(a) mAP of competence scores ($\mathcal{E}$ = cross-entropy) (b) mAP of competence scores ($\mathcal{E}$ = 0-1 error)

Figure 1: Competence Scores on Class Imbalanced CIFAR10

MODEL UNCERTAINTY

Table 1: mAP for Competence Prediction Under Model Uncertainty ($\mathcal{E}$ = cross-entropy). VGG16 is tested on CIFAR100 while the other models are on DIGITS. (U) is underfit, (W) is well trained, and (O) is overfit. Ablated ALICE refers to ALICE without the $p(\mathcal{E} < \delta | x, c_j)$ terms. Hyperparameters for these trials are in Appendix A.

Model	Accuracy	Softmax	TrustScore	Ablated ALICE	ALICE
MLP (U)	.121 ± .048	.0486 ± .015	.505 ± .27	.0538 ± .031	**.999 ± .0015**
MLP (W)	.898 ±.022	.989 ±.005	.929 ±.044	.958 ±.042	**.998 ±.001**
MLP (O)	.097 ±.015	.532 ±.062	.768 ±.064	.576 ±.033	**.996 ±.003**
RF (U)	.563 ± .078	.824 ± .16	.504 ± .33	.290 ± .322	**.999 ± .0011**
RF (W)	.930 ±.019	.998 ±.002	.898 ±.025	.923 ±.016	**.999 ±.000**
SVM (U)	.630 ±.018	.995 ±.003	.626 ±.046	.496 ±.069	**1.00 ±.000**
SVM (W)	.984 ±.009	**1.00 ±.000**	.931 ±.048	.963 ±.038	**1.00 ±.000**
SVM (O)	.258 ± .023	.200 ± .16	.215 ± .12	.252 ± .16	**.981 ± .028**
VGG16 (U)	.0878 ± .0076	.899 ± .014	.292 ± .049	.0369 ± .0041	**.913 ± .012**
VGG16 (W)	.498 ± .012	.975 ± .013	.604 ± .104	.0863 ± .0071	**.978 ± .0082**
VGG16 (O)	.282 ± .15	.659 ± .024	.665 ± .0080	.257 ± .018	**.738 ± .019**

DISTRIBUTIONAL UNCERTAINTY

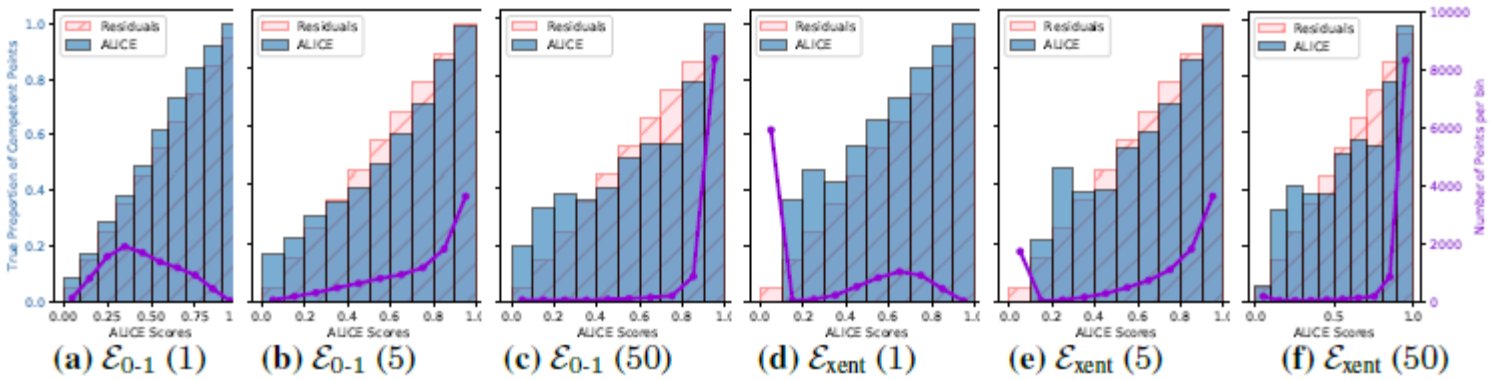
- ALICE can predict whether points are out of distribution, even without training on out of distribution data.

Table 2: mAP for Competence Prediction Under Distributional Uncertainty ($\mathcal{E} = \mathcal{E}_{\mathcal{D}}$).

CIFAR/SVHN Proportion	Softmax	TrustScore	Ablated ALICE	ALICE
10/90	.458 ±0.056	.518 ±0.039	.100 ±0.000	**.868 ±0.014**
30/70	.693 ±0.034	.721 ±0.026	.300 ±0.000	**.946 ±0.007**
50/50	.816 ±0.020	.833 ±0.015	.500 ±0.000	**.970 ±0.003**
70/30	.901 ±0.010	.910 ±0.008	.700 ±0.000	**.985 ±0.002**
90/10	.970 ±0.003	.972 ±0.002	.900 ±0.000	**.997 ±0.001**

CALIBRATION

- ALICE is calibrated at all stages of training and with all error functions tested.



(a) $\mathcal{E}_{0\text{-}1}$ (1) (b) $\mathcal{E}_{0\text{-}1}$ (5) (c) $\mathcal{E}_{0\text{-}1}$ (50) (d) $\mathcal{E}_{\text{xent}}$ (1) (e) $\mathcal{E}_{\text{xent}}$ (5) (f) $\mathcal{E}_{\text{xent}}$ (50)

Figure 3: ALICE score calibration of ResNet32 trained on CIFAR10, with various error functions and stages of training. The captions show the error functions and number of epochs trained.

CONCLUSION

- We introduce a generalized form of confidence called **competence**, that is dependent on the user's use-case for a machine learning model.
- ALICE is an accurate, calibrated competence estimator that doesn't require any out-of-distribution data and works on all classification models tested, both deep and classical.
- ALICE provides state-of-the-art results in common failure cases of competence estimators, such as class imbalance and out-of-distribution data.