Calibrating posterior estimates in NLP classifiers during training

Shirin Goshtasbpour , Rishabh Singh
Dept. of Computer Science
ETH Zurich
{shirin.goshtasbpour, risingh}@inf.ethz.ch

***Abstract*—Modern NLP classifiers are known to return uncalibrated estimations of class posteriors. Existing methods for posterior calibration rescale the predicted probabilities but often have little impact on final classifications and poorer generalization. We propose an end-to-end trained calibrator, Platt-Binning, that directly optimizes the objective while minimizing the difference between the predicted and empirical posterior probabilities. Our method leverages the sample efficiency of Platt scaling and the verification guarantees of histogram binning, thus not only reducing the calibration error but also improving task performance. Empirical evaluation on benchmark NLP classification tasks echo the efficacy of our proposal.**

I. Introduction

Deep learning has proven to be tremendously attractive for researchers in fields such as physics, biology, and manufacturing, to name a few [1], [2], [3]. However, these are fields in which representing model uncertainty is of crucial importance [4]. Deep Neural Networks are famous for providing highly accurate label predictions. Thus, a common way to incorporate them in other fields is to use the predictions of a trained classifier for decision making in a downstream task. In some cases the effectiveness of the decisions depends on a utility function and it is not enough to simply predict the most likely label for each example. What is needed instead is to quantify model uncertainty about the predictions. Despite promising performance in supervised learning benchmarks in terms of accuracy, Deep Neural Networks are poor at quantifying predictive uncertainty, and tend to produce overconfident predictions. Overconfident incorrect predictions can be harmful or offensive in NLP applications [5], hence proper uncertainty quantification is crucial in practice.

Probabilistic uncertainty in machine learning translates to estimation of probability mass function $p(y|\mathbf{x})$ by the model where $\mathbf{x}$ is the input sample and y is a class label. Recent efforts has also shown that state-of-the art structured prediction models are poorly calibrated. Therefore, blindly using the output of the softmax function output as the model uncertainty is misleading [6], [7], [8].

We are interested in calibrating the posterior estimates, i.e. we wish to get posterior probability estimations that reflect the true probability of the classes. If a model assigns 70% probability to an event, the event should occur 70% of the time if the model is calibrated. Even if the actual mechanism might be difficult to interpret, a calibrated model at least gives us a signal that it "knows what it doesn't know," thereby making these models easier to deploy in practice [9]. We define perfect calibration as follows.

$$\mathbb{P}(y|f(\mathbf{x})) = f(\mathbf{x})$$

where $f : \mathcal{X} \to \triangle_{K-1}$ is the probabilistic classifier that maps the samples to the K-dimensional simplex.

As majority of the current state-of-the art machine learning models, such as neural networks, do not output calibrated probabilities out of the box [10], existing works rely on re-calibration methods that take the output of an uncalibrated model, and transform it into a calibrated probability. One way of addressing this is to use Scaling approaches for re-calibration such as Platt scaling [11], isotonic regression [12], and temperature scaling [13]. These methods are widely used and require very few samples, however whether they actually produce calibrated probabilities is a topic of ongoing research. An alternative approach, histogram binning [14], outputs probabilities from a finite set. Histogram binning can produce a model that is calibrated, and unlike scaling methods we can measure its calibration error, but it is sample inefficient.In particular, the number of samples required to calibrate scales linearly with the number of classes the model needs to generate probability estimates for. Irrespective of the choice of the calibration method, existing works generally calibrate the posterior distribution predicted from the classifier after training. These post-processing calibration methods re-learn an appropriate distribution from a held-out validation set and then apply it to an unseen test set, causing a severe discrepancy in distributions across the data splits. The fixed split of the data sets makes the post-calibration very limited and static with respect to the classifier's performance.

In this paper we try to address some of the existing challenges in achieving apt calibration. In particular our contributions are:

- We propose a training technique that optimizes a binary classification objective for an NLP task by calibrating the posterior distribution while training.
- We leverage the advantages of both scaling and binning methods and propose a calibration method for NLP

classification task which is both sample efficient and verifiable.
- We investigate how the proposed method regularizes the performance of the classifier in question.

II. Literature Overview

Model uncertainty estimation and posterior calibration is a topic of continued interest not only in the fields of machine learning and statistics, but also in meteorology [15], fairness [16], healthcare [9], reinforcement learning [17], natural language processing [18], speech recognition [19] and economics [20]. In probabilistic models, the principal goal of estimation of the posterior $p(y|\mathbf{x})$ given a sample $\mathbf{x} \in \mathcal{X}$ and a label $y \in [K]$, is to assign low confidence to samples that were not explained well by the training data. One common way to calibrate multiclass posteriors after training the classifier $f : \mathcal{X} \to \mathbb{R}$ is to treat the problem as K one-vs-all binary problems. In this case, model uncertainty is quantified by normalizing the estimation of $p(y = k|f(\mathbf{x})_k)$ where $f(\mathbf{x})_k$ is the output score of the classifier for sample $\mathbf{x}$ and class k. Various binary calibration methods can be used to estimate the marginal posterior over a calibration dataset, ranging from parametric approaches (e.g. Platt scaling, temperature scaling, vector scaling [11], [13]), to non-parametric methods (e.g. quantile or bayesian binning [14], [21], and isotonic regression [12]. Another way to reduce the problem to binary calibration is by estimating model accuracy conditioned on its confidence, $p(y = \hat{y}|\max_{k\in[K]} f(\mathbf{x})_k)$.

Multiclass calibration aims to estimate the distribution of class labels conditioned on the estimated probability vector, $p(y|f(\mathbf{x}))$. In this case the sample complexity is exponential in the number of classes and therefore with large number of classes the main challenge is to constrain the hypothesis space with regularization. Some of the proposed methods for this purpose are matrix scaling and dirichlet scaling which both use linear models for estimation of $p(y = k|f(\mathbf{x}))$ [13], [22], and MLP and order preserving functions [23], [24].

Another approach is to account for model uncertainty via bayesian models. In Bayesian Neural Networks (BNNs) the predictive uncertainty will naturally be high in regions where training data is scarce [25]. However, the marginalization of the weights in BNN is intractable. Consequently, following papers propose various approximations such as variational inference (VI) [26], [27]. Although BNNs are theoretically proven to control the model overconfidence on unseen regions of data space [28], they require expensive approximations that limits their application in most modern NLP architectures. In [29] the authors model the distribution on posterior probability using a Dirichlet prior distribution and variational inference. MCDropout is a variational approximation of Gaussian processes that avoids explicit modeling of the posterior distribution [30]. Both of these methods require modification of training of the network.

In NLP tasks with structured outputs posterior calibration is particularly challenging since the number of classes are exponentially large and estimation of every posterior density or marginal posterior density is not possible. Previous works such as [31], [8] propose to use the downstream task with small number of classes to perform calibration and estimation of the calibration error. In structured prediction models calibration is also important for the generation of the structured outputs as the decoding algorithm relies on the posterior estimates to efficiently search through the space of sequences. However, estimation of the sequence calibration error and its correction is intractable. To cope with this problem, approximate calibration methods using a set of interesting events and feature based calibration are proposed in [32], [33] and an alternative calibration error estimator was proposed using sequence precision scoring function BLEU in [6].

Algorithm 1: Platt-Binning Calibrated Training

Input: Train set D, Bin B, Number of Classes K, Number of epochs e, Learning rate η, Number of updating empirical probabilities u
Output: Model Parameters Θ

Let Q : Empirical Probability Matrix $\in \mathbb{R}^{B\times K}$
Random initialization of Θ
for $i \in$ {*1,2,3, ...e*} **do**
  Break D into random mini-batches m
  for m *from* D **do**
    if *current step mod* $\lfloor e/u \rfloor == 0$ **then**
      $\hat{p}(x) = \max_k \text{softmax}(\Theta, D)_k \forall x \in D$
      $\hat{y} = \arg\max_k \text{softmax}(\Theta, D)_k \forall x \in D$
      $g = \arg\min_{g\in G} \sum_{(x,y)\in D} (\mathbb{1}_{y=\hat{y}} - g(\hat{p}(x)))^2$
      Choose bins so that an equal number of $g(p_i)$ in D land in each bin $b_j, j \in \{1, ...B\}$
      Discretize g, by outputting the average g value in each bin b_j: $\hat{g}_\beta(p_i) = \mu[\beta(g(p_i))]$
      $Q \leftarrow$ CalEmpProb($\hat{p}$;b_j)
    end
    $\Theta \leftarrow \Theta - \eta \nabla_\Theta L_{train}(\Theta, \hat{g}_\beta(p_i), b)$
  end
end

III. Our Method

Majority of the classification tasks in NLP work by first predicting a posterior probability distribution over all classes and then selecting the class with the largest estimated probability. However, these models are often poorly calibrated. Existing calibration methods re-learn an appropriate distribution from a held-out validation set and then apply it

to an unseen test set. The fixed split of the data sets and insufficient number of samples for training the calibration function adversely affects the generalization of post-hoc calibrated classifiers and reduces their accuracy.

Alternatively, we can dynamically estimate the required statistics for calibration from the train set during training iterations, thereby minimizing cross-entropy as well as the calibration error as a multi-task setup [31]. Given a training set $D = \{(x_1, y_1)..(x_n, y_n)\}$, where x_i is a n-dimensional vector of input features and y_i is a K-dimensional one-hot vector corresponding to its true label (with K classes), we minimize the loss L_{train}:

$$L_{train} = L_{class} + \lambda L_{cal} \tag{1}$$

Here L_{class} is the classification loss (for eg. cross-entropy) based on the predicted probability p_{ik} updated during training for sample i and class k:

$$L_{class} = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} log(p_{ik})$$

L_{cal} is the calibration loss which acts as a regularizer. It essentially tries to minimize the difference between the updated probability p and true posterior probabilities q via a distance function d (eg. mean squared error, KL-divergence, etc.):

$$L_{cal} = \sum_{i=1}^{N}\sum_{k=1}^{K} d(p_{ik}, q_{ik})$$

One crucial step here is to estimate the empirical probability q, which can be done by histogram binning method. Here, we measure the ratio of true labels for each bin split by the predicted posterior p from each update, and store in Empirical Probability Matrix $Q \in \mathbb{R}^{B\times K}$ where B is the number of bins used for each posterior dimension. Current methods update Q dynamically using histogram binning method. Histogram binning outputs probabilities from a finite set. It can produce a model that is calibrated, and unlike scaling methods we can measure its calibration error, but it is sample inefficient. In particular, the number of samples required to calibrate scales linearly with the number of distinct probabilities the model can output, B, which can be large particularly in the multi-class setting where B typically scales with the number of classes [34]. Recalibration sample efficiency is crucial- we often want to recalibrate our models in the presence of domain shift or recalibrate a model trained on simulated data, and may have access to only a small labeled dataset from the target domain. In this work, we propose to investigate how adaptive binning can circumvent this bottleneck. We leverage the sample efficiency of Platt scaling [11] and the verification guarantees of histogram binning [14] by defining the *Platt-Binning Calibrator*. The problem with scaling methods is we cannot estimate their calibration error. The upside of scaling methods is that if the function family has at least one function that can achieve calibration error ϵ they require $O(1/\epsilon^2)$ samples to reach calibration error ϵ, while histogram binning requires $O(B/\epsilon^2)$ samples. *Platt-Binning Calibrator* facilitates estimation of calibration error while being sample- efficient at the same time.

Since most modern deep learning classifiers do not output calibrated probabilities out of the box, recalibration methods take the output of an uncalibrated model, and transform it into a calibrated probability. That is, given a trained model $f : \mathcal{X} \to [0, 1]$, let $\mathbf{z} = f(\mathbf{x})$. We are given recalibration data $T = \{(\mathbf{z}_i, y_i)\}_{i=1}^{n}$ corresponding to model logits and the labels, and we wish to learn a calibrator $g : [0, 1] \to [0, 1]$ such that $g \circ f$ is well-calibrated. Conventional Scaling methods, for example Platt scaling, output a function g:

$$g = \arg\min_{g\in G} \sum_{(\mathbf{z},y)\in T} l(g(z), y)$$

where G is a the hypothesis class, $g \in G$ is differentiable, and l is a loss function, for example the log-loss or mean-squared error. The advantage of such methods is that they converge very quickly since they only fit a small number of parameters. On the other hand, Histogram binning constructs a set of bins that partitions $[0, 1]$ via a binning scheme. A binning scheme $\hat{B}$ of size B is a set of B intervals $I_1, ...I_B$ that partitions $[0, 1]$. Given $p = \text{softmax}(\mathbf{z})_k \in [0, 1]$, let $\beta(z) = j$, where j is the interval that p lands in $(p \in I_j)$. The binning scheme, $\hat{B}$ typically corresponds to choosing bins of equal widths (called equal width binning) or so that each bin contains an equal number of $\mathbf{z}_i$ values in the calibration dataset (called uniform mass binning). Histogram binning then outputs the average y_i value in each bin.

Platt-Binning Calibrator builds at the intersection of the above two methods. Given a recalibration data T of size n, *Platt-Binning Calibrator* outputs $\hat{g}_\beta$ such that $\hat{g}_\beta \circ f$ has a low calibration error by using the following procedure:
Step 1: Select $g = argmin_{g\in G} \sum_{(z,y)\in T}(y - g(z))^2$
Step 2: Choose the bins so that an equal number of $g(z_i)$ in T land in each bin I_j for each $j \in 1, ..., B$ —this uniform-mass binning scheme as opposed to equal-width binning is essential for being able to estimate the Expected Calibration Error, ECE.

$$\text{ECE} = \frac{1}{K}\sum_{k=1}^{K}\sum_{b=1}^{B}\frac{N_{kb}}{N_k}|Q_{bk} - \bar{p}_{bk}|$$

where $\bar{p}_{bk}$ is the average posterior estimate for class k for samples in bth bin and N_{kb} and N_k are the number of samples of class k assigned to bin b and in total, respectively.
Step 3: Discretize g, by outputting the average g value in each bin. Let $\mu(S) = \frac{1}{|S|}\sum_{s\in S} s$ denote the mean of a set of values S.We set $\hat{g}_\beta(z) = \mu(\beta(g(z)))$- we output the mean

value in the bin that $g(z)$ falls in.

The motivation behind our method is that the g values in each bin are in a narrower range than the label values y, so when we take the average we incur lower estimation error. If G is well chosen, our method requires $O(\frac{1}{\epsilon^2} + B)$samples to achieve calibration error ϵ instead of $O(\frac{B}{\epsilon^2})$ samples for histogram binning. All these steps are performed during training as explained in the pseudo-code below which makes this a novel calibrator to the best of our knowledge. In the following section we prove the efficacy of our method by carrying out extensive evaluation of the performance of pre-trained transformer models such as BERT [35] on simple multi-class text classification tasks. Our motivation comes from the analysis in [36] which shows that pre-trained models are significantly better calibrated when used out-of-the-box.

IV. Experiment

In the experiments we finetune the parameters on pre-trained BERT classifier using the regularized loss in equation (1). We compare our method to the following baselines:

- **MLE** is the baseline with maximum likelihood training without calibration where we simply report the results of vanilla BERT classifier on the chosen tasks.
- **Platt** scaling is a post-hoc calibration method where we calibrate the posterior estimations of MLE classifier using Platt scaling [11]. Formally, the parameters of the calibration functions $g(\mathbf{z}; \mathbf{W}, \mathbf{b}) = \sigma(\mathbf{W}\text{softmax}(\mathbf{z}) + \mathbf{b})$ is fit to the validation dataset where $\sigma(.)$ is the componentwise logistic function and fitting is performed using one-vs-all binarization of the classification task then instead of the estimated posterior $\text{softmax}(f(\mathbf{x}))_k$ for class k we return the calibrated value $g(f(\mathbf{x}))$ as the class probablity. Despite its simplicity this method is competitive with the more complex methods when implemented post-hoc [13].
- **PosCal** end-to-end training calibration using histogram binning [31]. In this method we have a nested training procedure where in the outer loop we fit a histogram binning scheme with fix widths to each dimension of the posterior estimates of the BERT model and we use Q_{bk} the ratio of samples of kth class that were assigned to bth bin as the empirical probability distribution q. In the inner loop we perform the ordinary training iterations over mini-batches of training dataset with cross-entropy loss and regularization term in equation (1) using KL-divergence between softmax output and the estimated empirical distribution.

$$L_{cal} = \sum_{i=1}^{N} \sum_{k=1}^{K} \log \frac{\text{softmax}(\mathbf{z}_i)_k}{Q_{\text{bin}(z_{ik})k}}$$

where $\text{bin}(.)$ returns the index of bin assigned to its input. In the experiments we used $\lambda = 1.0$, 10 bin for descretization of q and we update Q after every training epoch.

We test the baselines and our method on the benchmark on NLP classification tasks: xSLUE [37]. xSLUE contains classification benchmark on different types of styles such as a level of humor, formality and even demographics of authors. For the details we refer the reader to the original paper.

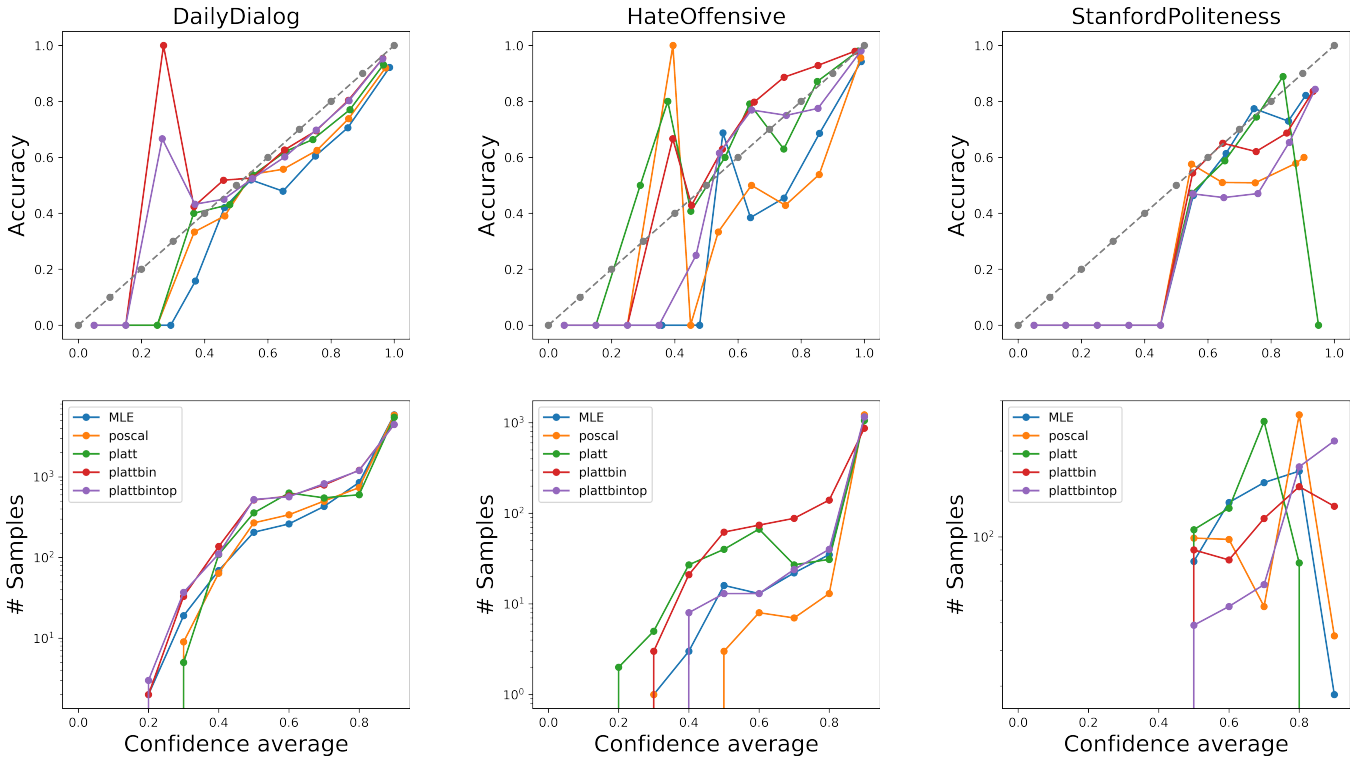
We setup the baseline experiments using the code in https://github.com/THEEJUNG/PosCal and to reproduce the results of [31]. We train to types of calibrators: in the first calibration task we train a calibrator for the most confident prediction of the classifier and call this version *plattbintop*. The pseudocode of this version is illustrated in algorithm (1). In the second version we train a separate Platt scaler and histogram binning for each class in a one-vs-all manner and we call this version of calibration *plattbin*. While this version is exactly the same as *plattbintop* for binary tasks, it results in a very different solution for tasks with $K > 2$. The pseudocode of this version is omitted due to being mainly similar to the other version with one additional loop over the classes at line 7 of algorithm (1) and conversion of label y and $\hat{y}$ to one-vs-all binary labels. We report task accuracy, F1 score and ECE. The results for MLE without calibration, PostCal, Platt-Binning for marginal probabilities, Platt-Binning for top predictions and Platt scaling on some of the tasks in xSLUE dataset are presented in the table I. The results are slightly different than what is reported in the paper.

V. Results and Discussion

Table I show task performance and calibration error on xSLUE benchmark datasets. In general, our method outperforms MLE and Poscal on approximately 50% of the datasets we tested in terms of both model performance and calibration error. For the rest of the dataset, our method gives competitive results. For instance, except *Trofi*, our method always generates lower calibration error that either of the other two baselines and in cases such as *DailyDialog*, *SentiTreeBank* and *ShortHumor*, the achieved reduction in ECE is significant. A key point to note here is that this reduction has not compromised with the model performance. For datasets which see a reduction in ECE compared to either of the baselines such as *DailyDialog*, *SarcasmGhosh*, *ShortHumor* and *StanfordPoliteness*, the accurcay and F1 score obtained by our model is at par with the rest two methods. In fact, cases like *SentiTreeBank* and *ShortRomance* even witness a significant improvement in the performance of the model when ECE is reduced. This observation proves the efficacy of our method in maintaining a perfect balance between model performance and model uncertainity- a testimony of an ideal calibrator. We refer the reader to Table VII in the

Dataset	Accuracy			F1 score			ECE		
	MLE	PosCal	Our	MLE	PosCal	Our	MLE	PosCal	Our
DailyDialog	84.8	**84.9**	84.1	29.4	**30.6**	29.8	16.5	13.2	**10.5**
HateOffensive	91.5	**94.4**	92.9	84.1	**86.5**	85.0	13.6	**8.3**	12.6
SarcasmGhosh	54.4	54.4	**54.5**	42.5	42.5	**43.0**	91.1	91.1	**89.9**
SentiTreeBank	94.6	93.9	**95.4**	94.6	93.9	**95.4**	9.6	8.0	**4.8**
ShortHumor	95.4	95.0	**95.7**	94.4	95.0	**95.7**	7.9	7.3	**5.9**
ShortRomance	99.0	96.0	**99.9**	98.9	95.9	**99.1**	**2.0**	7.1	4.3
StanfordPoliteness	**68.1**	56.1	67.9	**68.0**	53.5	67.9	**22.3**	59.1	23.0
TroFi	77.5	**78.8**	75.3	75.9	**77.7**	74.7	**18.4**	24.4	41.8
VUA	80.6	**81.6**	80.8	77.4	**78.5**	73.7	28.5	**14.7**	22.1

Table I: Comparison of Model performance and Calibration error on different benchmark datasets. Our method achieves better balance among the three metrics reported.

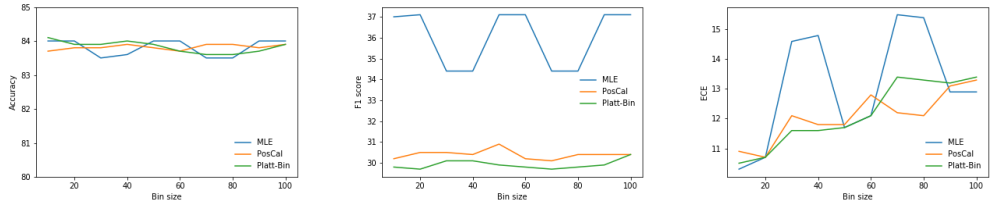


(a) (b) (c)

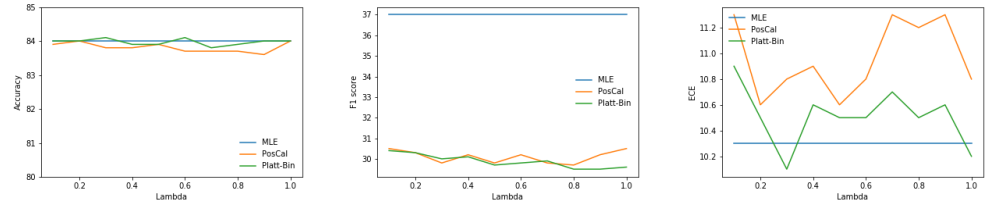
Figure 1: Calibration plots (top) accuracy-average confidence, (bottom) number of samples-average confidence

Appendix section. Our proposed method (PB or PBtop) is shown to achieve better results than baselines such as MLE, PosCal and post-hoc calibration using Platt Scaling.

We now analyse how our method behaves in comparison to MLE at sample level during test time. Table II shows a detailed analysis of misclassification made by MLE and Platt-Binning (PB). We see that both the methods have almost comparable performance in columns $A1$ and $A2$. As such, the number of samples for which MLE and PB gave different predictions (column M) is actually a small fraction of the total number of test samples used of evaluation of the methods (column $Test$). We further analyse the number



(a) Accuracy (b) F1 score (c) ECE



(d) Accuracy (e) F1 score (f) ECE

Figure 2: Effect of Bin-size (upper row) and regularization (lower row) on model performance and calibration error

Data	Test	M	P1	P2	A1	A2
DailyDialog	7740	475	244	192	84.7	84.1
HateOffensive	1255	93	32	**50**	91.4	**92.9**
SarcasmGhosh	2000	0	0	0	54.4	54.4
SentiTreeBank	1749	73	29	**44**	94.5	**95.4**
ShortHumor	2256	93	44	**49**	95.4	**95.6**
ShortRomance	100	0	0	0	99.9	99.9
StanfordPoliteness	567	75	38	37	68.1	67.9
TroFi	227	41	23	18	77.5	75.3
VUA	5873	958	472	**486**	80.6	**80.9**

Table II: Comparison of model performance at test time between MLE and PB. **Test**: Number of test samples, **M**: No. of test samples for which MLE and PLatt-Binning (PB) gave different predictions, **P1**: No. of samples correctly classified by MLE but misclassified by PB , **P2**: No. of samples correctly classified by PB but misclassified by MLE, **A1**: Accuracy of MLE, **A2**: Accuracy of PB

of samples where MLE gave correct predictions while PB failed to do so (column $P1$) and vice-versa (column $P2$). In 6 out of 9 dataset, PB demonstrates superior or similar performance (P2 $\geq$ P1). For rest of the three dataset, the difference is insignificant compared to the total size of the test set. This quantitative analysis reinstates that our method, PB, has better model performance at test time, thereby establishing that it generalizes well while reducing calibration error.

We extend the discussion above by analysing qualitative results in Table III. We consider three datasets- a two-class classification task *StanfordPoliteness*, a three-class classification task *HateOffensive* and a multi-class classification task ($K > 3$) *DailyDialog*, and include few test samples where MLE and PB disagreed on the predictions. The corresponding $\hat{p}$ along with the true label is also depicted.

In the first two cases from *StanfordPoliteness* dataset, the level of politeness (e.g., "Hey!" in S1) or arrogance (e.g., "What?" in S2) indicated on phrases is not captured well by MLE, so it predicts the incorrect label while PB gives a correct prediction. However, for the rest two cases, MLE gives confident correct predictions taking into account phrases such as "like" in S1 or a slightly difficult example in S2 but PB fails (only slightly in S2 though) to give correct predictions. Arguing on similar lines for the multi-class case, we witness cases where MLE fails to classify correctly (eg. S1 and S2 in *HateOffensive*) but PB gives highly confident predictions and vice-versa. From our manual investigation above, we find that statistical knowledge about posterior probability helps correct $\hat{p}$ while training PB, so making $\hat{p}$ switch its prediction. For further analysis, we provide more examples in the Appendix.

In Figure 1 we show the calibration plots for three datasets: *DailyDialog*, *HateOffensive*, and *StanfordPoliteness*. We divide test samples according to the most confident estimated posterior into 10 bins and we plot the accuracy of the classifier versus the average classification confidence in each one of the bins and the number of samples versus the classification confidence in the top and bottom row, respectively. Ideally, a calibrated classifier would assign a probability to the top class that is equivalent to its accuracy.

Data	Sentence	True Label	$\hat{p}$ (MLE)	$\hat{p}$ (PB)	MLE $\rightarrow$ PB
DailyDialog	**S1:** Really ? What did you get one for ?	surprise	0.17	0.60	INCOR$\rightarrow$COR
	S2: To hell with you . The accident was your fault	anger	0.14	0.41	INCOR$\rightarrow$COR
	S1: I might just ! Enjoy your stupid game !	anger	0.41	0.36	COR$\rightarrow$INCOR
	S2: Yeah . We rolled out the red carpet to welcome him home .	noemotion	0.96	0.37	COR$\rightarrow$INCOR
HateOffensive	**S1:** @HBergHattie @snkscoyote I wonder if the progs didn't relegate young black men to the ghettos to keep them away from harry reid's friends.	neither	0.02	0.91	INCOR$\rightarrow$COR
	S2: Every spic cop in #LosAngeles is loyal to the #LatinKin	hate	0.002	0.65	INCOR$\rightarrow$COR
	S1:"Our people". Now is the time for the Aryan race 2 stand up and say "no more". Before the mongerls turn the world into a ghetto slum.	hate	0.95	0.37	COR$\rightarrow$INCOR
	S2: #RebelScience .....is using an ACTUAL WOMAN as a genetic engineering lab for "all natural clones"..... or something..... #faggot #ro	hate	0.98	0.04	COR$\rightarrow$INCOR
StanfordPoliteness	**S1:** Hey, long time no seeing! How's stuff?	polite	0.16	0.63	INCOR$\rightarrow$COR
	S2: What user list? The one I linked to?	impolite	0.34	0.52	INCOR$\rightarrow$COR
	S1:I like the first shot. Are those doghouses?	polite	0.68	0.24	COR$\rightarrow$INCOR
	S2: I usually just boil water and then drink but I think it won't help here. Does it?	impolite	0.68	0.48	COR$\rightarrow$INCOR

Table III: Predicted $\hat{p}$ of true label from MLE and PB with corresponding sentences in D-Dialog, H-Offensive and S-Polite dataset. Provided examples contrast the predictions between MLE and PB for qualitative analysis.

Therefore, the accuracy-confidence curve of a calibrated classifier is close to the dashed grey curve in the top row. However, the distance of the curves is not enough to determine model calibration as most of the samples are assigned to the bin with highest estimated posterior. Thus correcting the calibration error in the bins with more samples is more effective in improving the expected calibration error. *Platt-Binning* and *Platt-Binning-Top* algorithms increase the number of samples with lower classification confidence in all three of the illustrated tasks, while in comparison to *MLE* with no regularization they only reduce classification accuracy by a negligible amount and even increases the accuracy for *HateOffensive* task. Although, the classifier become visibily underconfident in *HateOffensive* task where post-hoc Platt scaling has a more calibrated output. While the ECE doesn't improve in *StanfordPoliteness*, *Platt-Binning* algorithm doesn't increase the ECE as much as *PosCal* regularization. We conjecture this is happening due to better sample efficiency of this algorithm.

We conclude our analysis by observing the effect of two important parameters to this discussion- number of histogram bins used for calibration B and strength of the regularization, λ. Figure 2 shows how the key metrics- accuracy, F1 score and ECE of different methods vary when the number of bins B is varied as $\{10, ...100\}$. We see that the accuracy of all three methods- MLE, PosCal and PB does not change much and stay in the same range. Similarly the F1 score of PosCal and PB does not show much variation. It is easy to conclude that changing the bin size does not help with any improvements in the performance of the model. However, we see that the calibration error of all the methods have an increasing trend as B is increased. One plausible explanation can be that as we increase the number of bins, we don't have enough samples per bin to estimate the empirical probabilities accurately. Sine calibrated probabilities are used as an estimation of the true probabilities of the classes in case of PosCal and PB, it adds to the error if they are estimated wrongly. We now analyse the effect of regularization in Fig 2. Note that MLE assumes that there is no regularization, hence $\lambda = 0$ for MLE and we see a constant value for each case. The performance of PosCal and PB does not change much with increasing regularization. We see a lot of variation in ECE, however the trend is rather difficult to draw any conclusions about the effect of regularization.

VI. Conclusion

In this work we proposed a simple yet effective method called Platt-Binning calibrator for better posterior calibration. Our method has theoretically lower sample complexity than histogram binning, giving us the best of scaling and binning methods. And unlike the existing post-processing calibration methods, Platt-Binning directly penalizes the difference between the predicted and the true (empirical) posterior probabilities dynamically over the training steps thereby circumventing severe discrepancies in distributions

across data-splits. Our empirical analysis corroborates that Platt-Binning can not only reduce the calibration error but also increase the task performance on the classification benchmarks. Moreover, our method can be extended to any classification model as a form of regularization. There are many exciting avenues for future works in this regard. It will be interesting to assess how our method can provide advantages in the scenarios of domain adaptation and transfer learning. Moreover, exploring alternatives to the model family G from which estimate $\hat{g}$ is considered can be a direction of improvement. Lastly, optimizing the overall method for huge datasets can be an essential extension.

REFERENCES

[1] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature communications*, vol. 5, no. 1, pp. 1–9, 2014.

[2] O. Anjos, C. Iglesias, F. Peres, J. Martínez, Á. García, and J. Taboada, "Neural networks applied to discriminate botanical origin of honeys," *Food chemistry*, vol. 175, pp. 128–136, 2015.

[3] S. Bergmann, S. Stelzer, and S. Strassburger, "On the use of artificial neural networks in simulation-based manufacturing control," *Journal of Simulation*, vol. 8, no. 1, pp. 76–90, 2014.

[4] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," 2016.

[5] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016.

[6] A. Kumar and S. Sarawagi, "Calibration of encoder decoder models for neural machine translation," *arXiv preprint arXiv:1903.00802*, 2019.

[7] L. Dong, C. Quirk, and M. Lapata, "Confidence modeling for neural semantic parsing," *arXiv preprint arXiv:1805.04604*, 2018.

[8] K. Nguyen and B. O'Connor, "Posterior calibration and exploratory analysis for natural language processing models," *arXiv preprint arXiv:1508.05154*, 2015.

[9] X. Jiang, M. Osl, J. Kim, and L. Ohno-Machado, "Calibrating predictive model estimates to support personalized medicine," *Journal of the American Medical Informatics Association*, vol. 19, no. 2, pp. 263–274, 2012.

[10] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," *arXiv preprint arXiv:1807.00263*, 2018.

[11] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[12] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 694–699.

[13] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *arXiv preprint arXiv:1706.04599*, 2017.

[14] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in *Icml*, vol. 1. Citeseer, 2001, pp. 609–616.

[15] J. Bröcker, "Reliability, sufficiency, and the decomposition of proper scores," *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, vol. 135, no. 643, pp. 1512–1519, 2009.

[16] L. T. Liu, M. Simchowitz, and M. Hardt, "The implicit fairness criterion of unconstrained learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4051–4060.

[17] A. Malik, V. Kuleshov, J. Song, D. Nemer, H. Seymour, and S. Ermon, "Calibrated model-based deep reinforcement learning," *arXiv preprint arXiv:1906.08312*, 2019.

[18] D. Card and N. A. Smith, "The importance of calibration for estimating proportions from annotations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1636–1646.

[19] D. Yu, J. Li, and L. Deng, "Calibration of confidence measures in speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2461–2473, 2011.

[20] T. Gneiting, F. Balabdaoui, and A. E. Raftery, "Probabilistic forecasts, calibration and sharpness," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 2, pp. 243–268, 2007.

[21] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, vol. 2015. NIH Public Access, 2015, p. 2901.

[22] M. Kull, M. P. Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 316–12 326.

[23] A. Rahimi, K. Gupta, T. Ajanthan, T. Mensink, C. Sminchisescu, and R. Hartley, "Post-hoc calibration of neural networks," *arXiv preprint arXiv:2006.12807*, 2020.

[24] A. Rahimi, A. Shaban, C.-A. Cheng, B. Boots, and R. Hartley, "Intra order-preserving functions for calibration of multi-class neural networks," *arXiv preprint arXiv:2003.06820*, 2020.

[25] D. J. MacKay, "The evidence framework applied to classification networks," *Neural computation*, vol. 4, no. 5, pp. 720–736, 1992.

[26] A. Graves, "Practical variational inference for neural networks," in *Advances in neural information processing systems*, 2011, pp. 2348–2356.

[27] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.

[28] A. Kristiadi, M. Hein, and P. Hennig, "Being bayesian, even just a bit, fixes overconfidence in relu networks," *arXiv preprint arXiv:2002.10118*, 2020.

[29] T. Joo, U. Chung, and M.-G. Seo, "Being bayesian about categorical probability," *arXiv preprint arXiv:2002.07965*, 2020.

[30] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation," *arXiv preprint arXiv:1506.02157*, 2015.

[31] T. Jung, D. Kang, H. Cheng, L. Mentch, and T. Schaaf, "Posterior calibrated training on sentence classification tasks," *arXiv preprint arXiv:2004.14500*, 2020.

[32] V. Kuleshov and P. S. Liang, "Calibrated structured prediction," in *Advances in Neural Information Processing Systems*, 2015, pp. 3474–3482.

[33] A. Jagannatha and H. Yu, "Calibrating structured output predictors for natural language processing," *arXiv preprint arXiv:2004.04361*, 2020.

[34] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Binary classifier calibration: Non-parametric approach," *arXiv preprint arXiv:1401.3390*, 2014.

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[36] S. Desai and G. Durrett, "Calibration of pre-trained transformers," *arXiv preprint arXiv:2003.07892*, 2020.

[37] D. Kang and E. Hovy, "xslue: A benchmark and analysis platform for cross-style language understanding and evaluation," *arXiv preprint arXiv:1911.03663*, 2019.

APPENDIX

True Label	MLE → PB	MLE $\hat{p}$	PB $\hat{p}$	Sentence
happiness	INCOR→COR	0.32	0.70	Our pleasure . Please fill out this form , leaving your address and telephone number .
noemotion	INCOR→COR	0.30	0.55	sounds good . What are you going to have for your main course ?
surprise	INCOR→COR	0.17	0.60	Really ? What did you get one for ?
happiness	INCOR→COR	0.13	0.82	I'm glad to help you . What's wrong ?
anger	INCOR→COR	0.12	0.36	Damn it ! I'm injured here . We could wait all day for the police .
anger	INCOR→COR	0.14	0.41	To hell with you . The accident was your fault .
anger	INCOR→COR	0.11	0.39	To hell with you .
noemotion	COR→INCOR	0.73	0.43	No problem .
noemotion	COR→INCOR	0.99	0.31	Of course . The fitting room is right over there .
happiness	COR→INCOR	0.61	0.46	Great , thanks .
noemotion	COR→INCOR	0.78	0.34	Hello !
happiness	COR→INCOR	0.64	0.15	Sure thing , follow me . This here is the .
noemotion	COR→INCOR	0.90	0.36	Well , if you ever want to visit Korea , I would be happy to show you around .
anger	COR→INCOR	0.41	0.36	I might just ! Enjoy your stupid game !
noemotion	COR→INCOR	0.81	0.40	But he seems to be very happy with Rose .
happiness	COR→INCOR	0.53	0.08	So sorry . Next time we'll go , thanks anyway .
disgust	COR→INCOR	0.49	0.28	I dislike it most .
noemotion	COR→INCOR	0.98	0.42	It was a real red letter day for you .
noemotion	COR→INCOR	0.96	0.37	Yeah . We rolled out the red carpet to welcome him home .

Table IV: Additonal examples for predicted $\hat{p}$ of true label from MLE and PB with corresponding sentences in DailyDialog

True Label	MLE → PB	MLE $\hat{p}$	PB $\hat{p}$	Sentence
offensive	INCOR→COR	0.02	0.56	@aschops absolutely agree with that statement. It's just so amusing how angry it makes all these teabagger scumbags. That alone is worth i
neither	INCOR→COR	0.02	0.91	@HBergHattie @snkscoyote I wonder if the progs didn't relegate young black men to the ghettos to keep them away from harry reid's friends.
offensive	INCOR→COR	0.03	0.49	kieffer_jason i swear u a fuck nigga u a scary little bitch u think this a game hu
hate	INCOR→COR	0.32	0.60	@ImToBlame you a fatherless wallet carrying ass video game playing ass negro breh. You filth. No way you can afford to date a #TwitterHone
offensive	INCOR→COR	0.09	0.74	I hate a don't get shit done ass nigg
hate	INCOR→COR	0.002	0.65	Every spic cop in #LosAngeles is loyal to the #LatinKin
offensive	COR→INCOR	0.99	0.06	"@KingCuh: @16stanleys io io alu record ho vine sai pe hahahaha" lol anywaaaaaays..... ha
hate	COR→INCOR	0.98	0.04	#RebelScience .....is using an ACTUAL WOMAN as a genetic engineering lab for "all natural clones"..... or something..... #faggot #ro
offensive	COR→INCOR	0.99	0.38	"Let's do nips ahoy and spank me mayb
hate	COR→INCOR	0.95	0.37	"Our people". Now is the time for the Aryan race 2 stand up and say "no more". Before the mongerls turn the world into a ghetto slum. 14
offensive	COR→INCOR	0.68	0.47	😒RT @SedSince81: niggers RT @VonshayeB Before any moves are made... my black ass must take a na

Table V: Additonal examples for predicted $\hat{p}$ of true label from MLE and PB with corresponding sentences in HateOffensive

True Label	MLE → PB	MLE $\hat{p}$	PB $\hat{p}$	Sentence
impolite	INCOR→COR	0.34	0.52	What user list? The one I linked to?
polite	INCOR→COR	0.35	0.60	As I wrote above, at first I thought lets keep it, but after I heard some arguments, and when I made analysis of my own, I got to my conclusion. What's yours?
impolite	INCOR→COR	0.47	0.74	You and ¡url¿ are getting quite close to an edit war. Perhaps you should talk it out?
polite	INCOR→COR	0.16	0.63	Hey, long time no seeing! How's stuff?
polite	COR→INCOR	0.59	0.36	I am not sure of the question. Do you want problems that are obviously in one of the classes but not the other?
polite	COR→INCOR	0.62	0.45	092011 Try adding "ServerAlias mysite.com" after "ServerName" line. Also, do you have a DNS entry for mysite.com – same as www.mysite.com?
polite	COR→INCOR	0.68	0.24	I like the first shot. Are those doghouses?
impolite	COR→INCOR	0.51	0.44	Hmmm, Apple software on Windows question. I guess the "Apple Software" part defines the fact that you posted it here?
polite	COR→INCOR	0.61	0.49	how do you import the .csv into the spreadsheet? ('importdata'?)
impolite	COR→INCOR	0.68	0.48	I usually just boil water and then drink but I think it won't help here. Does it?
impolite	COR→INCOR	0.78	0.27	What's the benefit of the horizontal dropout? Is it safety? Is it just a style? Is it ease of maintenance?
impolite	COR→INCOR	0.51	0.32	Maybe it's necessary to phrase this another way: is there any food that *everybody* can eat?

Table VI: Additonal examples for predicted $\hat{p}$ of true label from MLE and PB with corresponding sentences in StanfordPoliteness

	Accuracy					F1 score					ECE				
Dataset	MLE	PosCal	PBtop	posPS	PB	MLE	PosCal	PBtop	posPS	PB	MLE	PosCal	PBtop	posPS	PB
DailyDialog	84.8	**84.9**	83.7	84.8	84.1	29.4	**30.6**	29.9	28.4	29.8	16.5	13.2	11.5	**9.6**	10.5
HateOffensive	91.5	94.4	**95.9**	93.4	92.9	84.1	6.5	**91**	86.8	85.0	13.6	8.3	**3.8**	3.9	12.6
SarcasmGhosh	54.4	54.4	54.5	54.4	**54.5**	42.5	42.5	42.6	42.5	**43.0**	91.1	91.1	90.9	**89.7**	89.9
SentiTreeBank	94.6	93.9	**95.8**	94.5	95.4	94.6	93.9	**95.8**	94.5	95.4	9.6	8.0	5.1	7.1	**4.8**
ShortHumor	95.4	95.0	**95.8**	95.5	95.7	94.4	95.0	**95.8**	95.5	95.7	7.9	7.3	4.6	**3.6**	5.9
ShortRomance	99.0	96.0	98	99	**99.9**	98.9	95.9	97.9	98.9	**99.1**	**2.0**	7.1	4.5	**2.0**	4.3
StanfordPoliteness	**68.1**	56.1	66.8	67.9	67.9	**68.0**	53.5	65.6	66.9	67.9	22.3	59.1	34.4	**8.1**	23.0
TroFi	77.5	**78.8**	74	77.5	75.3	75.9	**77.7**	73.5	76.2	74.7	18.4	24.4	43.6	**16.7**	41.8
VUA	80.6	**81.6**	81.3	81.2	80.8	77.4	**78.5**	74.6	77.5	73.7	28.5	14.7	19.9	**6.5**	22.1

Table VII: Comparison of Model performance and Calibration error on different benchmark datasets. MLE: Maximum Likelihood; PosCal: Posterior Calibrated Training with Histogram Binning; posPS: post-hoc calibration with Platt scaling; PB: Platt-Binning Method; PBtop: PB over max(softmax(logits)). Our method (PB or PBtop) achieves better balance among the three metrics reported.