Fill In The Gaps: Model Calibration and Generalization with Synthetic Data

Yang Ba[1], Michelle V. Mancenido[2], and Rong Pan[1]

[1]School of Computing and Augmented Intelligence, Arizona State University
[2]School of Mathematical and Natural Sciences, Arizona State University
[1]yangba@asu.edu, Rong.Pan@asu.edu, [2]mmanceni@asu.edu

Abstract

As machine learning models continue to swiftly advance, calibrating their performance has become a major concern prior to practical and widespread implementation. Most existing calibration methods often negatively impact model accuracy due to the lack of diversity of validation data, resulting in reduced generalizability. To address this, we propose a calibration method that incorporates synthetic data without compromising accuracy. We derive the expected calibration error (ECE) bound using the Probably Approximately Correct (PAC) learning framework. Large language models (LLMs), known for their ability to mimic real data and generate text with mixed class labels, are utilized as a synthetic data generation strategy to lower the ECE bound and improve model accuracy on real test data. Additionally, we propose data generation mechanisms for efficient calibration. Testing our method on four different natural language processing tasks, we observed an average up to 34% increase in accuracy and 33% decrease in ECE.

1 Introduction

Natural Language Processing (NLP) models have fundamentally advanced the syntactic and semantic analysis, information retrieval, and automated generation of textual data. State-of-the-art (SOTA) models (e.g., transformers (Vaswani et al., 2017), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019)) have excelled in practical, user-centric applications such as automated customer support chatbots, personalized content curation, and real-time multilingual text translation. Other NLP models, which are typically trained for a specialized use context, have also been developed and fine-tuned for numerous downstream tasks, including sentiment analysis, named entity recognition (NER), and text classification, as parts of a decision-support system (DSS). Powered by deep learning algorithms, these classification models have achieved remarkable levels of performance in terms of their accuracy, F1 scores, and AUCs (Li et al., 2020; Cohan et al., 2019).

As machine learning philosophies continue to evolve, growing attention is placed on metrics beyond simple classification accuracy. In recent years, socially responsible artificial intelligence (AI) has been strongly advocated by algorithmic regulatory frameworks (e.g., the US Algorithmic Accountability Act (Donovan et al., 2018)), especially in safety-critical domains, such as healthcare (Pfohl et al., 2022) and law enforcement (Salvador et al., 2021). Some key pillars of socially responsible AI include *accountability*, *transparency*, and *robustness* (Cooper et al., 2022). Ensuring a calibrated ML model accountable for its decision means that it must provide clear justifications for any decision being made, while transparency requires that these justifications are understandable and interpretable (Kadavath et al., 2022); additionally, robustness requires that the ML model performs consistently well under various conditions. In classification tasks, these requirements can be addressed by properly managing model output uncertainty, i.e., quantifying, calibrating, and communicating the proper confidence level associated with each prediction to the end user. Among the three aspects of uncertainty management, calibration directly improves model performance by ensuring that model predictions are congruent with empirically observed outcomes.

AI risk management is an emerging field that emphasizes understanding the limitations of model predictions. Model calibration techniques are used to address the fact that high accuracy does not always mean high confidence in a model's predictions. For example, consider a classifier trained to recognize handwritten digits. This model might achieve high accuracy on a test set, but it also provides the predicted probability for each class, which reflects its level of uncertainty. If it classi-

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 17211–17225

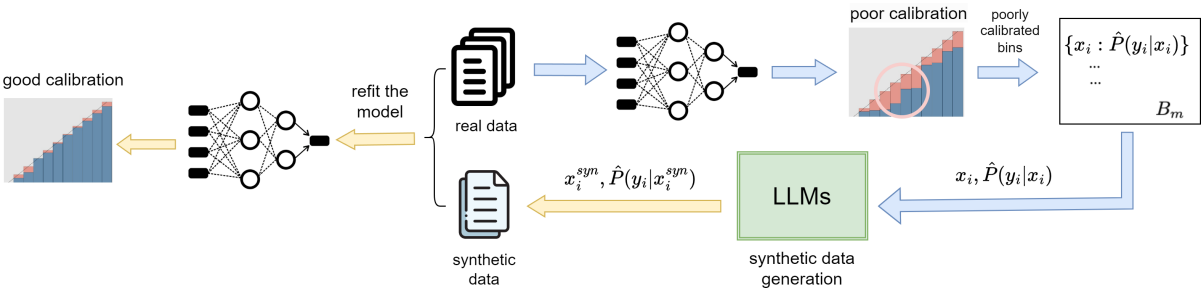


Figure 1: The framework of our proposed method involves initially training a downstream model using real data to identify poorly calibrated bins. Data instances, x_i, and their prediction probabilities, $\hat{P}(y_i|x_i)$, from these bins are then fed into large language models (LLMs) to generate synthetic data, x_i^{syn}, along with their corresponding probabilities, $\hat{P}(y_i|x_i^{syn})$. This synthetic data, combined with the real data, is used to retrain the downstream model, thereby improving calibration outputs without compromising model performance.

fies a digit as a '3' with 70% probability and as an '8' with 30% probability, it indicates that while the model predicts '3', it lacks high confidence. Understanding this prediction uncertainty has several key benefits: (1) refining decision-making thresholds to improve overall model performance; (2) adjusting models to perform well under different conditions and data distributions; (3) reducing the black-box nature of machine learning models, fostering greater transparency and trust; and (4) enabling more consistent and reliable decision-making, particularly in risk-sensitive applications where errors can have significant consequences.

Modern deep learning neural networks (NN), however, have been shown to be often miscalibrated i.e., while the NN model performs well in classification, the uncertainty around predictions is also high (Wang et al., 2021; Minderer et al., 2021). NLP models trained on classification tasks (such as sentiment analysis) are built on deep learning algorithms, with many hidden layers and regularization steps, and consequently, numerous hyperparameters to be tuned. Recent work has shown the association between increased depth and/or width of NN layers, and miscalibrated outcomes (Guo et al., 2017). Model calibration becomes worse in data-scarce scenarios, where the fraction of events predicted does not align with actual outcomes because the amount of data available at hand may not be sufficient enough to be representative across different classes. Data augmentation approaches (e.g., the mixup approach (Zhang et al., 2018; Thulasidasan et al., 2019)) and the associated model calibration problems have been discussed in literature (Wen et al., 2021). But, theories for understanding the association between model performance and calibration are still lacking.

This work is motivated by a recently published paper (Sahu et al., 2023), in which LLMs are utilized to generate synthetic data close to the decision boundary to sharpen the discrimination power of the classifier and increase model accuracy. The LLM-generated synthetic data leverage the capability of LLMs in providing both realistic and diverse datasets, which potentially increases the ML model's generalizability on out-of-distribution data. The application of synthetic data has been explored as an augmented training set (Van Breugel et al., 2023), validation set (Shoshan et al., 2023), or test set (van Breugel et al., 2023) to improve ML model performance. However, applying synthetic data to address model calibration has barely been explored. We aim to use LLM-generated synthetic text data to fine-tune downstream binary classification tasks to reduce expected calibration error (ECE) without sacrificing the ML model's accuracy.

Our approach is derived from the Probably Approximately Correct (PAC) learning framework (Valiant, 1984). We prove that reducing both calibration and misclassification errors can be achieved simultaneously, and we establish the necessity of generating synthetic data for enhancing model calibration. This approach is validated on real-world text datasets. The synthetic data generation process is accomplished using open-source Large Language Models - Llama 2 (Touvron et al., 2023). Figure 1 illustrates our proposed framework to improve model calibration and generalization via synthetic data for natural language classification tasks.

The contributions of this work can be summarized as follows:

1. We derive the Expected Calibration Error bound to explore the possibility of achieving

both high accuracy and low ECE.

2. We propose a strategy for fixing calibration errors and filling the gaps in the reliability diagram.

3. We demonstrate the effectiveness of purposefully augmenting LLM-generated synthetic data into the training set to achieve ML model prediction uncertainty calibration.

2 Calibration Concept

This section introduces some basic concepts related to model calibration, which would lay a foundation to derive our methodology.

Expected Calibration Error (ECE). ECE is a widely-used metric to evaluate how well a model's predicted probabilities (confidence) align with its actual outcomes (accuracy) (Guo et al., 2017). It is calculated by segmenting the full range of predicted probabilities into M equal bins and sorting predictions into these bins based on their confidence. Within each bin, the model's accuracy (the fraction of correct predictions) and average predicted confidence are computed. Let B_m be the set of examples in the m^{th} bin, whose accuracy and confidence are:

$$\begin{aligned} \text{Acc}(B_m) &= \frac{1}{|B_m|} \sum_{x_i \in B_m} 1(\hat{y}_i = y_i), \\ \text{Conf}(B_m) &= \frac{1}{|B_m|} \sum_{x_i \in B_m} \hat{p}_i \end{aligned} \tag{1}$$

where $1(\hat{y}_i = y_i)$ is an indicator function that is equal to 1 if $\hat{y}_i = y_i$ and 0 otherwise; $\hat{p}_i$ is the predicted probability associated with the instance x_i. The concept of accuracy here is based on each class of labels, which is a subset of the well-accepted model evaluation metric: *accuracy*. The Expected Calibration Error (ECE) given n examples is defined by taking a weighted average of the absolute differences between the bin's confidence and its accuracy:

$$\text{ECE} = \textstyle\sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Acc}(B_m) - \text{Conf}(B_m)| \tag{2}$$

There exist some variants of ECE, like MCE (Guo et al., 2017), ACE(Nixon et al., 2019), and other metrics to quantify calibration like brier score (Rufibach, 2010), however, in this paper, the calibration error refers to ECE.

Reliability Diagram. Reliability diagram (see Figure 2a) is a tool to visualize the model calibration. $Conf(B_m)$ and $Acc(B_m)$ represent the x-axis and y-axis of the diagram respectively for bin B_m. The diagonal line denotes perfectly calibrated and any deviations from this diagonal line indicate a model's miscalibration. Therefore, the miscalibration can be divided as above the line (underconfidence: $Acc(B_m) > Conf(B_m)$) and under the line (overconfidence: $Acc(B_m) < Conf(B_m)$) areas. We use a reliability diagram to find out the target bins where synthetic data are needed to fill in.

3 Methodology

In this section, we utilize the Probably Approximately Correct (PAC) learning framework to derive the expected calibration error(ECE) bound and discuss the benefits of using synthetic data to improve models' calibration and generalization. Moreover, we use a toy sample to demonstrate our methodology.

3.1 From PAC Learning to Expected Calibration Error Bound

Probably Approximately Correct (PAC) learning (Valiant, 1984) offers a theoretical framework that establishes the bounds on learning model parameters with specified levels of error and confidence, relating model accuracy to confidence level and sample size. According to Hoeffding's inequality,

$$P(|E(h) - E(h^*)| > \epsilon) \leq 2\exp(-2\epsilon^2 n)$$

where $E(h)$ denotes the true error of the hypothesis h on unseen data and $E(h^*)$ denotes the empirical error of the hypothesis h on the training data. Let δ [1] be the confidence level and make $\delta = 2\exp(-2\epsilon_a^2 n)$. This inequality presents the maximum allowable difference between the true and empirical errors based on a given sample size n and desired uncertainty level δ. Thus, we get the minimal sample size to make the hypothesis true considering error difference ϵ and confidence δ is given by $n = \log(2/\delta)/(2\epsilon^2)$.

Now we derive the ECE bound from the PAC learning framework. First, we extend the definition of accuracy and confidence in Equation (1) from

[1]Confidence level in PAC learning refers to a probability that the learned hypothesis with an error rate less than a specified accuracy; confidence in ECE represents the predicted probability for a given prediction.

bin-wise to data-wise. Then we have

$$\text{Acc}(X) = \sum_{m=1}^{M} \frac{|B_m|}{n} \text{Acc}(B_m),$$
$$\text{Conf}(X) = \sum_{m=1}^{M} \frac{|B_m|}{n} \text{Conf}(B_m)$$

The dataset is denoted by $\{X, y\}_i^n$, where X denotes the feature space, $X \in \{X_1, \ldots, X_n\}$ and y is the label $y \in \{y_1, \ldots, y_n\}$. Based on Hoeffding's inequality, we have

$$P(|Acc(X) - Acc(X^*)| > \epsilon_a) \leq 2\exp(-2\epsilon_a^2 n) \quad (3)$$

where $Acc(X)$ denotes the expected accuracy in the model and $Acc(X^*)$ is the observed accuracy of training data. ϵ_a is the error for accuracy and we let $\delta_a = 2\exp(-2\epsilon_a^2 n)$.

Proposition – Expected Calibration Error Bound. *Given n training samples, if the probability of the difference between the expected model parameter and its estimated value being less than ϵ_a is at least $(1 - \delta_a)\%$, then the probability of the difference between the expected calibration error and the estimated calibration error in the training samples being less than ϵ_{ECE} is at least $(1 - \delta_{ECE})\%$. Here, δ_{ECE} =2δ_a, and $\epsilon_{ECE} = \epsilon_a + |Conf(X) - Conf(X^*)| = \epsilon_a + \sum_{m=1}^{M} \frac{|B_m|}{n} |Conf(B_m) - Conf(B_m^*)|$.*

A shortened proof (detailed proof is provided in Appendix B) is given below:

By deriving from the left side of equation (3), we get:

$$\begin{aligned} &P\left(|\text{Acc}(X) - \text{Acc}(X^*)| > \epsilon_a\right) \\ \geq &P\left(\text{ECE}(X) - \text{ECE}(X^*) > \epsilon_a + |\text{Conf}(X) - \text{Conf}(X^*)|\right) \end{aligned} \quad (4)$$

Combined with the right side of equation (3):

$$\begin{aligned} &P\left(\text{ECE}(X) - \text{ECE}(X^*) > \epsilon_a + |\text{Conf}(X) - \text{Conf}(X^*)|\right) \\ &\leq 2\exp(-2\epsilon_a^2 n) \\ \Rightarrow &P\left(|\text{ECE}(X) - \text{ECE}(X^*)| > \epsilon_{ECE}\right) \\ &\leq 4\exp(-2(\epsilon_{ECE} - |\text{Conf}(X) - \text{Conf}(X^*)|)^2 n) \end{aligned} \quad (5)$$

where $\epsilon_{ECE} = \epsilon_a + |\text{Conf}(X) - \text{Conf}(X^*)| = \epsilon_a + \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Conf}(B_m) - \text{Conf}(B_m^*)|$. And $n_{ECE} = \log(4/\delta_{ECE})/(2(\epsilon_{ECE} - |\text{Conf}(X) - \text{Conf}(X^*)|)^2)$.

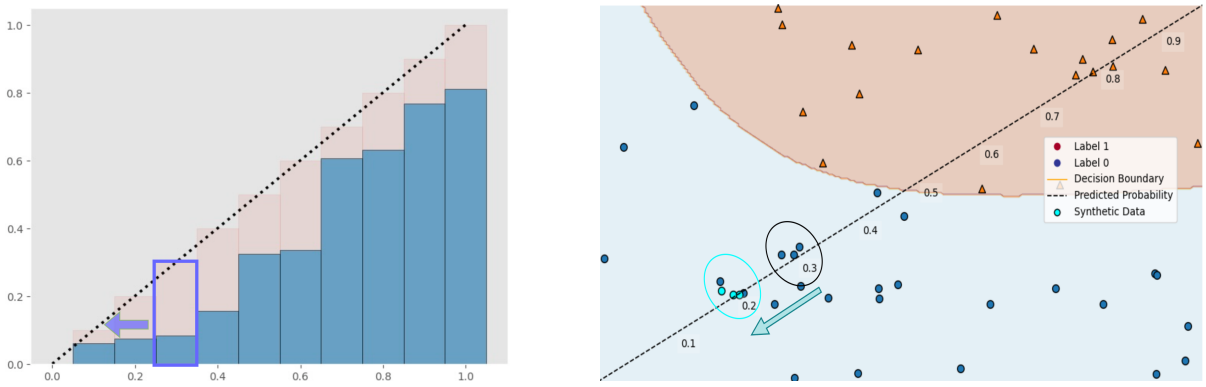
Hoeffding's inequality holds on Bernoulli random variables and accuracy is computed by counting correct predictions. By introducing the concept of uncertainty δ, we obtain the relationship among error, uncertainty, and sample size for PAC learning. Then we can derive the ECE bound from the same inequality and ECE is a random variable with a value in [0, 1]. Finally, given a sample size, we have the relationship among ϵ_a,ϵ_{ECE},δ_a, δ_{ECE}.

Remark 1: *Since the difference between true prediction probabilities and the estimated prediction probabilities exists, given the same data points to train a model, the error for ECE is larger than the error for accuracy compared with the true metrics and the uncertainty level for ECE is two times that for accuracy.*

Remark 2: *Based on $n_{ECE} = \log(4/\delta_{ECE})$ / $(2(\epsilon_{ECE} - |Conf(X) - Conf(X^*)|)^2)$, increasing the amount of training data will result in smaller error ϵ_{ECE} and lower uncertainty level δ_{ECE}; similar to the effects on ϵ_a and δ_a.*

Remark 3: *Since $\epsilon_{ECE} = \epsilon_a + |Conf(X) - Conf(X^*)|$, reducing $|Conf(X) - Conf(X^*)|$ aligns the ECE error with the accuracy error. This alignment helps models achieve both good calibration and better generalization.*

Remark 1 explains why some neural networks own a good performance but are more likely to be ill-calibrated. *Remark 2* indicates that increasing the amount of training data can both improve model generalization and lower expected calibration error. When the training data is insufficient, synthetic data is the natural option to augment the training data size. *Remark 3* provides insights into what kind of synthetic data is needed to fix the calibration issue. The newly added synthetic data should reduce the difference between predicted probabilities and the true probabilities. We cannot know the true difference of $|\text{Conf}(X) - \text{Conf}(X^*)|$, but we can decrease it by reducing $|\text{Conf}(B_m) - \text{Conf}(B_m^*)|$ in bins that display the gaps given that $\text{Conf}(X) = \sum_{m=1}^{M} \frac{|B_m|}{n} \text{Conf}(B_m)$, as we know where the perfect calibration line is for each bin. Therefore, we can manipulate the prediction probability of synthetic data to minimize the difference. In other words, synthetic data is applied to fill the gaps against the perfect calibration. That is, we try to decrease ECE by using synthetic data to lower the ECE bound.



(a) Reliability diagram with the target bin

(b) Move generated synthetic data away from DB

Figure 2: Generating synthetic data to address miscalibration gaps. In (a), the target bin for calibration is identified as *Low Probability & Overconfidence*. Synthetic data is generated away from the decision boundary in (b).

3.2 Synthetic Data Generation Strategy

Synthetic data generation consists of two stages: First, we specify the gaps against the perfect calibration line in the reliability diagram. Bins over the line are underconfident while those under the line are overconfident. The data points in those bins are the target data samples for synthetic data generation. With the predicted probability 0.5 as the cutoff, we categorize the reliability diagram (Figure 2a) into four scenarios: *Low Probability & Over Confidence, Low Probability & Under Confidence, High Probability & Over Confidence, and High Probability & Under Confidence*, as shown in Table 1.

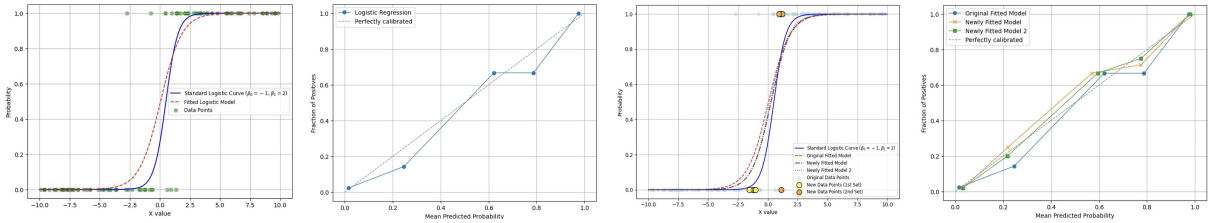
Next, LLMs, which serve as text generators, are used to create synthetic text data. Since LLMs are trained on diverse and extensive data spanning a wide range of sources, we can distill the knowledge from LLMs to generate synthetic data that is considered out-of-distribution of training data. We ask LLMs to imitate the classifier we trained by generating similar instances using data samples we collected from the target bin. Specifically, we pass the data instance x_i and $\hat{P}(y_i|x_i)$ from a trained classifier to LLMs and ask it to generate a similar instance x_i^{syn} with $\hat{P}(y_i|x_i^{syn})$, where $|\hat{P}(y_i|x_i)$ - $\hat{P}(y_i|x_i^{syn})|$ = $|\text{Conf}(B_m) - \text{Conf}(B_m^*)|$. For example, suppose there are n_{bins} bins, if the target text is from m^{th} bin and $|\text{Conf}(B_m) - \text{Conf}(B_m^*)|$ is α, then we will ask LLMs to generate the synthetic texts that have the $\frac{m}{n_{bins}} \pm \alpha$ probability belonging to one class and the $1 - (\frac{m}{n_{bins}} \pm \alpha)$ probability for the other class.

To illustrate our method, in Figure 2b a hidden predicted probability line is shown orthogonal to the estimated decision boundary. Data points close to this decision boundary would be predicted with around 0.5 probability (the softmax output), while data points at the two ends of this line could be predicted with close to 0.1 or 0.9 probability, respectively. Suppose that our targeted bin has a confidence of 0.3 and it is over-confident as shown in Figure 2a (highlighted in a purple rectangular). The gap between the empirical and theoretical uncertainty values is shown in red. There are two possible solutions to fill the gap: 1) increasing the number of incorrect predictions, thus raising the blue bar that represents the empirical inaccurate prediction percentage; or 2) moving this bin to the left, into the bin with a smaller uncertainty value. We use the second solution to align the miscalibration bins because the first solution could harm the accuracy of the classifier.

In Figure 2b, the target data points are circled in black and the synthetic data are in the blue circle, which are generated based on the generation strategy in Table 1. Since the synthetic data shares a similar feature space and the same labels as the target data samples, the retrained classifier would predict them as the same class but with smaller probabilities. This makes it more likely for the synthetic data to be assigned to the same bin as the original target data samples. In this way, we push

	Over Confidence	Under Confidence
Low Probability ($\hat{P}(y_i\|x_i) \leq 0.5$)	Decrease predicted prob (Move away from DB)	Increase predicted prob (Move towards DB)
High Probability ($\hat{P}(y_i\|x_i) > 0.5$)	Increase predicted prob (Move towards DB)	Increase predicted prob (Move away from DB)

Table 1: Synthetic Data Generation Strategy (DB: Decision Boundary). Refer to Appendix A for the prompts used for data generation across different scenarios.



(a) (b) (c) (d)

Figure 3: The iterative process of enhancing the accuracy and calibration of a 1D logistic regression model is demonstrated. Initially, the model is fitted using observed data (a), followed by the creation of its reliability diagram to identify poorly calibrated bins (b). Next, synthetic data points are strategically added to two targeted bins and the model is refitted. This iterative approach results in the model closely approximating the true logistic curve (c), thereby improving the calibration in the reliability diagram (d).

the predicted probability of this bin away from the decision boundary and reduce the difference $|\text{Conf}(B_m) - \text{Conf}(B_m^*)|$.

Utilizing LLM, we employ a two-stage approach to ensure both the fidelity and diversity of the synthetic data generated. We obtain the instance x_i^{syn} with probability $\hat{P}(y_i^{syn}|x_i^{syn})$ in the first stage and relabel it via LLMs to ensure it belongs to the same class of x_i in the second stage. Since step 1 mixes up information from two labels to some degree, it enhances data diversity. The relabeling of the second stage confirms that the generated texts belong to the correct label, which guarantees its fidelity.

3.3 Toy Example

We use a 1D logistic regression classifier as an example to demonstrate that adding appropriate synthetic data in the target bins can produce a better-calibrated and more accurate model. Parameters of the true model are defined: β_0 = -1 and β_1 = 2. We randomly simulate 300 data points from the range between -10 and 10 and classify them based on the true model as the label. A logistic regression model is fitted on these data points. The fitted parameters are β_0 = -0.06 and β_1 = 1.13. The model achieves an accuracy of 0.95 and an ECE of 0.0405 (Figure 3a).

Figure 3b shows us that the fitted logistic regression is overconfident about its predictions in the 2^{nd} bin and 4^{th} bin. Now we target these two bins to generate some synthetic data points to fill the gap. The function we used to generate synthetic data points is a left-sided truncated normal distribution, whose parameters are: $\mu = \mu_{Bin_i}$, $SD = SD_{Bin_i}$, n = $|Bin_i|$, $i = \{2, 4\}$. We add new data points step by step to see how the logistic curve changes: 1) add synthetic data of the 2^{nd} bin, 2) then add synthetic data of the 4^{th} bin based on previously added data points. See the parameters and performance for newly fitted models below:

- orginal data (**orginal fitted model**): β_0 = -0.06 and β_1 = 1.13, ACC: 0.95, ECE: 0.0405;
- synthetic data in 2^{nd} bin (**newly fitted model**): β_0 = -0.339 and β_1 = 1.2627, ACC: 0.95327, ECE: 0.0424;
- synthetic data in 2^{nd} bin + 4^{th} bin (**newly fitted model 2**): β_0 = -0.2558 and β_1 = 1.2953, ACC: 0.9469, ECE: 0.0366.

Figure 3c and Figure 3d illustrate that incorporating synthetic data generated from overconfident bins shifts the fitted logistic curve towards the standard logistic curve, resulting in a more accurately calibrated model.

Dataset	Classes	Balanced	Class Prop.	#Train/#Test
TC	2	No	65:35	3104/345
SUBJ	2	Yes	50:50	8000/2000
B77	2	Yes	44:56	177/80
SE	2	No	35:65	3012/750
Arxiv	2	Yes	50:50	4800/1200
Medical	2	Yes	50:50	2662/1060

Table 2: Dataset Statistics

4 Experiment

4.1 Dataset

We employ four datasets on text classification tasks across multiple domains with varying sample sizes and proportions of class. To better evaluate our approach, we select two balanced and two imbalanced datasets, respectively, and the sample size varies from hundreds to thousands. The Complaints dataset (TC) (Preoţiuc-Pietro et al.,

Dataset	Input		Prompt	Generated Text (x_i^{syn} , $\hat{P}(y_i\|x_i^{syn})$)	Label (Y)
SE	x_i	The zoom function on this camera is so loud that sometimes you will be unable to use it if you find yourself in a situation where you must be quiet.	An example x_i which belongs 75% to negative and 25% to positive (based on a classifier's categorization). Now I ask you to act as that classifier and based on this example, generate a diverse set of 3 short utterances where each utterance belongs 55% to negative and 45% to positive.	I love how this router can handle a large network, but the price is a bit steep for my taste. (55% negative, 45% positive)	negative
	$\hat{P}(y_i\|x_i)$	0.75			
	miscalibration type	overconfidence			

Table 3: An example of generating synthetic data via LLM. As an example, we use the SE dataset; for information on other datasets, see Table 8 in Appendix D. **Input** contains the original text (x_i) and the average predictive probability of the bin it comes from ($\hat{P}(y_i|x_i)$). **Generated Text** is the one after the relabeling process. **Note**: during re-fine tuning of the downstream model, we exclude $\hat{P}(y_i|x_i^{syn})$ – (55% negative, 45% positive) and retain only x_i^{syn} in the dataset.

2019) contains 3K tweets regarding product reviews, which are categorized as complaints or not-a-complaints. The SUBJectivity dataset (SUBJ) (Pang and Lee, 2004) is a benchmark dataset that contains 10K objective/subjective movie reviews. Banking77 (B77) (Casanueva et al., 2020) is a dataset comprising fine-grained intents within the banking domain featuring multiple classes. For our study, we select instances from two of these classes, which makes the size of the dataset relatively small. SentEval (SE) (Hu and Liu, 2004) contains 3K data used for sentiment analysis tasks.

Additionally, to discover whether the pre-trained knowledge of LLMs is a crucial element in determining the performance of generated synthetic data on downstream tasks. We choose two newly-released datasets that are unlikely to be a part of the training data of LLMs we used in the paper (*Llama-2*): Arxiv-10 (Farhangi et al., 2022) and Medical (Fansi Tchango et al., 2022). We pick 'cs' and 'stat' classes, and randomly sample 30% from the entire data in Arxiv-10. Its task is to classify the subject based on the title of a paper. Medical is the dataset in a medical diagnosis domain categorizing a specific disease based on a patient's symptoms, where "Influenza" and "Anaphylaxis" in our experiment. See detailed statistics of these datasets in Table 2.

4.2 Training

All datasets are split into training, validation, and test sets. For the TC dataset, we split the entire data into training, validation, and test sets with a ratio of 80:10:10. For other datasets, the validation sets are created by randomly sampling 20% from the training set. All experiments are evaluated on their original test sets. We fine-tune BERT$_{base}$ (Devlin et al., 2019) models for text classification by adding a dropout and softmax layer following the pre-trained structure. We train each model 5 epochs and apply a 1e-6 learning rate and 50% dropout.

Step 1. After completing the training process, we calculate the reliability diagram and the difference (D) between the proportion of positive labels and the mean predicted values for each bin based on the validation set. If the absolute value of D_m in m^{th} bin is larger than the threshold 0.03, the data in the m^{th} bin will be selected to generate synthetic data.

Step 2. We use LLMs to generate synthetic data based on the texts in the target bin from Step 1. To explore the effect of the number of bins (M), we select three scenarios by setting M = 10, 15, 20.

4.3 Synthetic Data Generation

Synthetic text generation is performed using version *Llama-2-7b-chat-hf* of Llama 2 at a temperature $T = 0.1$. We apply the two-stage and three-shot learning generation method proposed in the paper (Sahu et al., 2023) to guarantee diversity and authenticity. First, we define each label and provide three examples for each one (Appendix C). Then, we present the example text from the previous selection stage along with the predicted probability of this example that was extracted from the trained BERT$_{base}$ classifier. We then instruct llama 2 to act as the base classifier to generate three similar texts that could be classified with specific probability requirements. Next, we instruct llama 2

	TC		SUBJ		B77		SE		Arxiv		Medical	
Metric	ACC	ECE	ACC	ECE	ACC	ECE	ACC	ECE	ACC	ECE	ACC	ECE
Baseline	0.867 (0.00)	0.058 (0.02)	0.955 (0.01)	0.034 (0.01)	0.708 (0.12)	0.234 (0.04)	0.884 (0.01)	0.06 (0.00)	0.805 (0.00)	0.105 (0.01)	0.864 (0.00)	0.051 (0.01)
Isotonic	0.871 (0.00)	0.082 (0.01)	0.959 (0.00)	0.027 (0.01)	0.850 (0.02)	0.063 (0.01)	0.890 (0.01)	0.058 (0.01)	0.812 (0.01)	0.114 (0.01)	0.869 (0.01)	0.069 (0.01)
Platt scaling	0.863 (0.01)	0.086 (0.01)	0.955 (0.01)	0.029 (0.00)	0.846 (0.03)	0.207 (0.03)	0.888 (0.01)	0.068 (0.00)	0.807 (0.01)	0.122 (0.00)	0.869 (0.01)	0.065 (0.01)
MC dropout	0.868 (0.02)	0.054 (0.01)	0.952 (0.01)	0.032 (0.01)	0.821 (0.23)	0.274 (0.14)	0.876 (0.01)	0.050 (0.02)	0.799 (0.01)	0.058 (0.04)	0.871 (0.01)	0.070 (0.01)
Temp scaling	0.867 (0.01)	0.049 (0.01)	0.955 (0.01)	0.026 (0.01)	0.708 (0.12)	0.253 (0.17)	0.884 (0.01)	0.038 (0.00)	0.805 (0.00)	0.070 (0.01)	0.864 (0.00)	0.056 (0.01)
10 bins												
Synthesis	0.867 (0.01)	0.053 (0.01)	0.960 (0.01)	0.027 (0.01)	0.625 (0.07)	0.255 (0.10)	0.871 (0.00)	0.055 (0.02)	0.815 (0.01)	0.077 (0.03)	0.873 (0.01)	0.048 (0.01)
Synthesis+	0.886 (0.01)	0.046 (0.01)	0.961 (0.00)	0.03 (0.00)	0.792 (0.20)	0.231 (0.03)	0.889 (0.01)	0.064 (0.00)	0.808 (0.01)	0.099 (0.01)	**0.871 (0.00)**	**0.047 (0.01)**
15 bins												
Synthesis	0.879 (0.01)	0.049 (0.01)	0.961 (0.00)	0.026 (0.00)	0.800 (0.11)	0.224 (0.08)	**0.904 (0.00)**	**0.04 (0.00)**	0.802 (0.00)	0.096 (0.01)	0.875 (0.00)	0.052 (0.00)
Synthesis+	0.881 (0.01)	0.050 (0.01)	**0.9605 (0.00)**	**0.024 (0.00)**	0.863 (0.09)	0.203 (0.10)	0.901 (0.01)	0.055 (0.01)	0.824 (0.01)	0.087 (0.01)	0.879 (0.01)	0.055 (0.01)
20 bins												
Synthesis	0.883 (0.00)	0.046 (0.01)	0.959 (0.00)	0.027 (0.00)	0.808 (0.12)	0.180 (0.07)	0.900 (0.00)	0.048 (0.00)	0.818 (0.01)	0.089 (0.01)	0.871 (0.01)	0.054 (0.00)
Synthesis+	**0.890 (0.00)**	**0.046 (0.01)**	0.959 (0.00)	0.026 (0.00)	**0.950 (0.04)**	**0.224 (0.03)**	0.896 (0.01)	0.049 (0.01)	**0.820 (0.00)**	**0.075 (0.00)**	0.867 (0.01)	0.046 (0.01)

Table 4: Model Performance and Calibration on Real Test Data. Highlighted values considered both ACC and ECE and weigh more on ECE.

to relabel the generated texts to ensure them belong to the "correct" class. Table 3 illustrates the inputs, prompts, and outputs for generating synthetic data using the SE dataset. Additional prompts for different scenarios can be found in Appendix D.

4.4 Evaluation

Results are assessed on real test set. **Baseline** refers to the results trained on the model in Step 1. Suppose we have a total of N original data points in the training and validation set, and there are S_1 data points in target bins from the validation set. Let LLMs generate S_2 synthetic data points, and $S_2 = S_1$. **Synthesis** refers to the results that we retrain the model by replacing S_1 original data points with S_2 synthetic data points. **Synthesis+** indicates that we add S_2 synthetic samples into the original N data points.

In addition to the baseline, we also compare the performance of our methods against some widely used model calibration techniques. **Isotonic regression** (Zadrozny and Elkan, 2001) employs a non-parametric method that adjusts predicted probabilities to align with observed outcomes, and **Platt scaling** (Platt et al., 1999) fits a logistic regression model to calibrate classifier scores based on predicted probabilities. **Monte Carlo dropout** (Gal and Ghahramani, 2016) randomly masks nodes to estimate the probability distribution. In our paper, the model makes 10 predictions for each instance and each time with a different dropout mask. **Temperature scaling** (Guo et al., 2017) works by dividing the pre-softmax output by a temperature T and the optimal value of T is estimated by the validation dataset. All experiments use the same BERT$_{base}$ model parameters.

4.5 Results

We run each experiment for three random seeds and report the average value (with standard deviation in brackets) of accuracy and ECE in Table 4. By adding synthetic data with a size of 7%-18% [2] of the training set, we would have a 21-33% ECE decrease. Taking both accuracy and ECE into account, our synthetic data replacement (synthesis) and synthetic data add-on (synthesis+) methods outperform other calibration approaches in five out of six datasets. Temperature scaling can sometimes achieve lower ECE, but a key disadvantage is that it doesn't affect accuracy. On the other hand, while dropout can improve model calibration, it carries the risk of reducing accuracy, as seen in the results on the Arxiv-10 dataset. We also observe a

[2]The validation set is used to identify poorly calibrated bins. We set a predefined threshold of 0.03, and only bins with gaps exceeding this threshold are selected.

clear association of a larger bin number with lower ECE[3]. In addition, the results of our approach have smaller variances compared with those of the baseline.

Whether imbalanced datasets (TC and SE) or balanced datasets (SUBJ, B77, Arxiv, and Medical), improvements in uncertainty calibration are fairly comparable on average. It is also shown that even though the baseline model for the SUBJ dataset already has outstanding accuracy, our approach can still make the model better calibrated without degrading the model's classification performance. Results from B77 have a larger variance due to its smaller data size.

4.6 Ablation Study

To discuss if the LLM's self-calibration capability strongly impacts our approach, we instruct *Llama-2-7b-chat-hf* with few-shot learning and set top_k = 1 in which we obtain the conditional probability of one class $P(label|text)$. Then we computer the accuracy and ECE from LLMs.

	LLM_{ACC}	LLM_{ECE}	$\text{Syn}_{ACC}(\%)$	$\text{Syn}_{ECE}(\%)$
LLM_{ACC}	1	-0.737	0.592	-0.566
LLM_{ECE}	-0.737	1	-0.026	0.423

Table 5: Pearson Correlation Coefficient on six datasets. $\text{Syn}_{ACC}(\%)$ and $\text{Syn}_{ECE}(\%)$ denotes the percentage of the downstream model's accuracy increases and how much percentage of the expected calibration error is decreased respectively.

In Table 5, we didn't observe a strong negative correlation coefficient between LLM_{ACC} and $\text{Syn}_{ECE}(\%)$, indicating there is no empirical evidence that shows the calibration ability of LLMs determines the application of our proposed methods. Additionally, we found a moderate positive association between the llama's accuracy and the accuracy improvement in downstream tasks. This suggests that the prediction accuracy of LLMs, rather than calibration capability, plays a more important role in downstream models' performance. Therefore, using advanced LLMs (such as Llama 3.2) or fine-tuning LLMs to incorporate domain knowledge could yield better performance when applying our approach.

[3]We use the bin's average confidence to represent each instance within that bin, so having more bins could lead to more accurate probability estimates in synthetic data generation.

5 Related Work

Model calibration has emerged as an open challenge in machine learning as concerns arise regarding the responsible and ethical use of ML-enabled systems. Several methods have been proposed, including Platt Scaling (Platt et al., 1999), Isotonic regression (Zadrozny and Elkan, 2001), among others. Both of them somewhat change the predicted probability, which could lower the predicted accuracy. In the computer vision field, a mixup method (Zhang et al., 2018) has been proposed to overcome the shortcomings of data scarcity. It combines two instances from the original dataset with different proportions. A follow-up paper (Wen et al., 2021) investigates the computer vision task calibration by using the mixup approach and concludes that this approach could impair the model calibration. However, the calibration issue in the NLP field has rarely been discussed.

Several benefits from using synthetic data have been explored in (Sahu et al., 2023). It is found that ML prediction accuracy can be improved significantly by adding synthetic data generated near the decision boundary. On the other hand, a recent paper (Li et al., 2023) investigates on potential and limitations of synthetic data generated from LLM for text classification tasks and concludes that while synthetic data can be beneficial in certain scenarios, it does not consistently enhance model performance. Our research is different from theirs in that we provide a strategy that enables both good generalization and uncertainty calibration.

6 Conclusion

In the era of large models, we believe smaller models still hold tremendous values in, e.g., edge computing and specialized downstream machine learning tasks. We derive the expected calibration error bound for ML models and explore the possibility of leveraging synthetic data to mitigate calibration error. Through empirical validation with text classification tasks, we demonstrate the usefulness of our method; that is, by harnessing the power of LLMs, purposefully generated synthetic data can be utilized to train smaller downstream NLP tasks, achieving both strong classification performance and calibration error reduction.

Limitations and Future Work

While increasing the sample size generally helps reduce the Expected Calibration Error (ECE), sim-

ply adding more synthetic data may not always lead to optimal model performance, as excessive synthetic data can cause overfitting. Therefore, the focus should be on generating high-quality data and strategically identifying instances that require better calibration. It's also important to note that the primary objective of this paper is not to compare or evaluate the capabilities of Large Language Models (LLMs). Rather, we assume that an updated, optimally performing LLM could generate higher-quality synthetic data, which could, in turn, enhance the accuracy of downstream tasks and improve model calibration using our proposed methodology.

In our experiments, we applied a 0.03 threshold to filter out ill-calibrated bins, leaving room for future work to investigate how varying cutoff values might influence calibration enhancement. While our method focuses on text classification applications, there is potential to extend this approach to other downstream NLP tasks. Additionally, future research could explore the use of generative models beyond Large Language Models (LLMs), broadening the scope of applicability. Finally, extending our method to multi-class classification models is proposed as an area for future work.

References

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. 2019. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3693–3699, Hong Kong, China. Association for Computational Linguistics.

A Feder Cooper, Emanuel Moss, Benjamin Laufer, and Helen Nissenbaum. 2022. Accountability in an algorithmic society: relationality, responsibility, and robustness in machine learning. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 864–876.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Joan Donovan, Robyn Caplan, Jeanna Matthews, and Lauren Hanson. 2018. Algorithmic accountability: A primer.

Arsene Fansi Tchango, Rishab Goel, Zhi Wen, Julien Martel, and Joumana Ghosn. 2022. Ddxplus: A new dataset for automatic medical diagnosis. *Advances in neural information processing systems*, 35:31306–31318.

Ashkan Farhangi, Ning Sui, Nan Hua, Haiyan Bai, Arthur Huang, and Zhishan Guo. 2022. Protoformer: Embedding prototypes for transformers. In *Advances in Knowledge Discovery and Data Mining: 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16–19, 2022, Proceedings, Part I*, pages 447–458.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *arXiv preprint arXiv:2004.00584*.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34:15682–15694.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019. Measuring calibration in deep learning. In *CVPR workshops*, volume 2.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Stephen Pfohl, Yizhe Xu, Agata Foryciarz, Nikolaos Ignatiadis, Julian Genkins, and Nigam Shah. 2022. Net benefit, calibration, threshold selection, and training objectives for algorithmic fairness in healthcare. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1039–1052.

John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large-Margin Classifiers*, 10(3):61–74.

Daniel Preoţiuc-Pietro, Mihaela Gaman, and Nikolaos Aletras. 2019. Automatically identifying complaints in social media. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5019, Florence, Italy. Association for Computational Linguistics.

Kaspar Rufibach. 2010. Use of brier score to assess binary predictions. *Journal of Clinical Epidemiology*, 63(8):938–939.

Gaurav Sahu, Olga Vechtomova, Dzmitry Bahdanau, and Issam Laradji. 2023. PromptMix: A class boundary augmentation method for large language model distillation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5316–5327, Singapore. Association for Computational Linguistics.

Tiago Salvador, Stephanie Cairns, Vikram Voleti, Noah Marshall, and Adam Oberman. 2021. Faircal: Fairness calibration for face verification. *arXiv preprint arXiv:2106.03761*.

Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, Matan Fintz, and Gérard Medioni. 2023. Synthetic data for model selection. In *International Conference on Machine Learning*, pages 31633–31656. PMLR.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in Neural Information Processing Systems*, 32.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Leslie G Valiant. 1984. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

Boris Van Breugel, Zhaozhi Qian, and Mihaela Van Der Schaar. 2023. Synthetic data, real errors: how (not) to publish and use synthetic data. In *International Conference on Machine Learning*, pages 34793–34808. PMLR.

Boris van Breugel, Nabeel Seedat, Fergus Imrie, and Mihaela van der Schaar. 2023. Can you rely on your model evaluation? Improving model evaluation with synthetic test data. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. 2021. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820.

Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. 2021. Combining ensembles and data augmentation can harm your calibration. In *International Conference on Learning Representations*.

Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 609–616. ACM.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

A Appendix

Our code is implemented based on Pytorch 2.2.1 and the pre-trained $\text{Bert}_{\text{base}}$ model is downloaded from the huggingface library. Both llama2 and $\text{Bert}_{\text{base}}$ run on Nvidia P100 GPUs.

	Parameters
optimizer	Adam
max length	512
embedding dim	768
batch size	32
learning rate	1e-6
dropout	0.5
epoch	5

Table 6: Model Parameters ($\text{Bert}_{\text{base}}$)

We provide the code used to generate the corresponding prompts based on the scenarios to which the bins in Table 1 belong.

```
def gen_prompt(conf, diff, label0, label1,
    indicator):
    """ diff: the gap against the perfect
    calibration line.
        indicator: based on table 1, which the
    bin belongs to.
        revised_conf: set a bound if the gap is
    too large
        that makes the generated instances
    assigned correct labels. """

    if 'low' in indicator and 'under' in
    indicator:
        revised_conf = 45 if conf + diff >= 50
    else conf + diff
        generation_prompt = f"which belongs
    {100-conf}% to {label0} and {conf}% to {
    label1}
        (based on a classifier's categorization).
     Now I ask you to act as that classifier
        and based on this example, generate a
    diverse set of 3 short utterances where
    each
        utterance belongs {100-revised_conf}% to
     {label0} and {revised_conf}% to {label1}:
    "

    if 'low' in indicator and 'over' in
    indicator:
        revised_conf = 5 if conf - diff <= 0
    else conf - diff
        generation_prompt = f"which belongs
    {100-conf}% to {label0} and {conf}% to {
    label1}
        (based on a classifier's categorization).
     Now I ask you to act as that classifier
        and based on this example, generate a
    diverse set of 3 short utterances where
    each
        utterance belongs {100-revised_conf}% to
     {label0} and {revised_conf}% to {label1}
        (no explanation):"

    if 'high' in indicator and 'under' in
    indicator:
        revised_conf = 95 if conf + diff >= 100
    else conf + diff
        generation_prompt = f"which belongs
    {100-conf}% to {label0} and {conf}% to {
    label1}
        (based on a classifier's categorization).
     Now I ask you to act as that classifier
        and based on this example, generate a
    diverse set of 3 short utterances where
    each
        utterance belongs {100-revised_conf}% to
     {label0} and {revised_conf}% to {label1}
        (no explanation):"

    if 'high' in indicator and 'over' in
    indicator:
        revised_conf = 55 if conf - diff <= 50
    else conf - diff
        generation_prompt = f"which belongs
    {100-conf}% to {label0} and {conf}% to {
    label1}
        (based on a classifier's categorization).
     Now I ask you to act as that classifier
        and based on this example, generate a
    diverse set of 3 short utterances where
    each
        utterance belongs {100-revised_conf}% to
     {label0} and {revised_conf}% to {label1}
        (no explanation):"

    return generation_prompt
```

B Appendix

The Expected Calibration Bound Proof:
From equation (1) in section 2, we extend the definition of accuracy and confidence from bin-wise to data-wise:

$$\text{Acc}(X) = \sum_{m=1}^{M} \frac{|B_m|}{n} \text{Acc}(B_m), \quad \text{Conf}(X) = \sum_{m=1}^{M} \frac{|B_m|}{n} \text{Conf}(B_m)$$

correspondingly,

$$\text{Acc}(X^*) = \sum_{m=1}^{M} \frac{|B_m^*|}{n} \text{Acc}(B_m^*), \quad \text{Conf}(X^*) = \sum_{m=1}^{M} \frac{|B_m^*|}{n} \text{Conf}(B_m^*)$$

According to Hoeffding's inequality, we have:

$$P(|Acc(X) - Acc(X^*)| > \epsilon_a) \leq 2\exp(-2\epsilon_a^2 n).$$

where, $Acc(X)$ means the expected accuracy in the model; $Acc(X^*)$ is the observed accuracy of training data. ϵ_a is the error for accuracy and we let $\delta_a = 2\exp(-2\epsilon_a^2 n)$. The derivation from the left side of the inequality:

$$\begin{aligned}
&P\left(|\text{Acc}(X) - \text{Acc}(X^*)| > \epsilon_a\right) \\
=&P\left(\left|\sum_{m=1}^{M} \frac{|B_m|}{n} \text{Acc}(B_m) - \sum_{m=1}^{M} \frac{|B_m|}{n} \text{Acc}(B_m^*)\right| > \epsilon_a\right) \\
=&P\left(\sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Acc}(B_m) - \text{Conf}(B_m) + \text{Conf}(B_m) + \text{Conf}(B_m^*) - \text{Conf}(B_m^*) - \text{Acc}(B_m^*)| > \epsilon_a\right) \\
=&P\left(\sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Acc}(B_m) - \text{Conf}(B_m) - [\text{Acc}(B_m^*) - \text{Conf}(B_m^*)] + \text{Conf}(B_m) - \text{Conf}(B_m^*)| > \epsilon_a\right) \\
\geq&P\left(\sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Acc}(B_m) - \text{Conf}(B_m) - [\text{Acc}(B_m^*) - \text{Conf}(B_m^*)]| - \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Conf}(B_m) - \text{Conf}(B_m^*)| > \epsilon_a\right) \\
=&P\left(\sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Acc}(B_m) - \text{Conf}(B_m) - [\text{Acc}(B_m^*) - \text{Conf}(B_m^*)]| > \epsilon_a + \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Conf}(B_m) - \text{Conf}(B_m^*)|\right) \\
\geq&P\left(\sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Acc}(B_m) - \text{Conf}(B_m)| - \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Acc}(B_m^*) - \text{Conf}(B_m^*)| > \epsilon_a + |\text{Conf}(X) - \text{Conf}(X^*)|\right) \\
=&P\left(\text{ECE}(X) - \text{ECE}(X^*) > \epsilon_a + |\text{Conf}(X) - \text{Conf}(X^*)|\right)
\end{aligned}$$

Combined with the right side of the inequality:

$$\begin{aligned}
&P\left(\text{ECE}(X) - \text{ECE}(X^*) > \epsilon_a + |\text{Conf}(X) - \text{Conf}(X^*)|\right) \leq 2\exp(-2\epsilon_a^2 n) \\
=&P\left(|\text{ECE}(X) - \text{ECE}(X^*)| > \epsilon_a + |\text{Conf}(X) - \text{Conf}(X^*)|\right) \leq 4\exp(-2\epsilon_a^2 n) \\
=&P\left(|\text{ECE}(X) - \text{ECE}(X^*)| > \epsilon_{ECE}\right) \leq 4\exp(-2\epsilon_a^2 n) \\
=&P\left(|\text{ECE}(X) - \text{ECE}(X^*)| > \epsilon_{ECE}\right) \leq 4\exp(-2(\epsilon_{ECE} - |\text{Conf}(X) - \text{Conf}(X^*)|)^2 n)
\end{aligned}$$

where $\epsilon_{ECE} = \epsilon_a + |\text{Conf}(X) - \text{Conf}(X^*)| = \epsilon_a + \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{Conf}(B_m) - \text{Conf}(B_m^*)|$.

C Appendix

	System Prompt
TC	Consider the task of classifying between the following classes (along with some examples): 1. complaint, which is about customer inquiries on a state of affairs, product, organization or event to express a negative mismatch between reality and expectations. Some examples of utterances include: - Dear @nvidia, I don't think I should have to roll back to driver v270.61 to make my games work, and my desktop not glitch out. - @FC_Help hi m order is 913181 did you revise the money? if you did.. how about the shipping ? - @FC_Help Will you be getting the wendy cotton v neck dress in pavlova back in stock on the site? 2. not_complaint, which is the opposite of complaint mentioned above, about customer regular or normal inquiries on a state of affairs, product, organization or event without any expression related to a negative mismatch between reality and expectations. Some examples of utterances include: - @FC_Help How can I get a hold of you so we can discuss the problem I am having with my coat? - @FC_Help I need to check my order. - @FC_Help looking for "bright carol" or "stained glass" dress. do you have these in stock anymore?
SUBJ	Consider the task of classifying between the following classes (along with some examples): 1. objective, which is assigned to text that presents factual information, descriptions, or statements without personal opinions, emotions, or bias. It focuses on delivering facts or information that is independent of the writer's personal feelings or beliefs. Some examples of utterances include: - "nicklas passes out , and the next day when he returns to school he notices that nobody seems to notice him." - "when reuben buys a black-market cure for his unusual chest complaint, jenny is forced to make a terrible sacrifice." - "raj has always had a unrequited childhood crush on a friend named tina, but tina's best friend pooja has always had a crush on raj." 2. subjective, which is applied to text that expresses personal opinions, feelings, beliefs, or thoughts. It often includes evaluative language, personal experiences, or interpretations, reflecting the writer's personal stance or emotional reaction. Some examples of utterances include: - "for its seriousness, high literary aspirations and stunning acting, the film can only be applauded." - "an inelegant combination of two unrelated shorts that falls far short of the director's previous work in terms of both thematic content and narrative strength." - "what's needed so badly but what is virtually absent here is either a saving dark humor or the feel of poetic tragedy."
B77	Consider the task of classifying between the following classes (along with some examples): 1. age_limit, which is about customer inquiries on age-related restrictions for opening a bank account. Some examples of utterances include: - Can I get an account for my son? - Can my teenager have an account? - How young can I be to open my own account? 2. atm_support, which is about users asking how to use an ATM, where to find one, or any other clarifications about a transaction at an ATM. Some examples of utterances include: - Is the closest ATM to me within 2 miles? - Are there only certain ATM machines where I can use this card? - Do you know the closest ATM?
SE	Consider the task of classifying between the following classes (along with some examples): 1. negative, which is assigned to content that expresses negative feelings, emotions, or attitudes. Examples include statements of dissatisfaction, sadness, anger, or criticism. Some examples of utterances include: - "even with newborn diapers it filled way too fast." - "bluetooth does not work on this phone." - "also, some other mp3 players such as the nitrus allow you to play wma (windows media audio) files, whereas the ipod does not." 2. positive, which is applied to content that expresses positive feelings, emotions, or attitudes. Examples include statements of happiness, satisfaction, praise, or optimism. Some examples of utterances include: - "4 megapixels is enough for anybody and the photo quality is awesome." - "an extra gig of room, fm radio, voice recorder, pim applications with sync to outlook." - "all the buttons & necessary lil' gadgets are on the sides of the player which is nice for when you are holding it in the palm of your hand."

Table 7: System Prompt for Data Generation

D Appendix

Dataset	Input		Prompt	Generated Text (x_i^{syn} , $\hat{P}(y_i\|x_i^{syn})$)	Label (Y)
TC	x_i	@UbisoftSupport When will u guys fix the jager glitch?	An example x_i which belongs 75% to complaint and 25% to not_complaint (based on a classifier's categorization). Now I ask you act as that classifier and based on this example, generate a diverse set of 3 short utterances where each utterance belongs 65% to complaint and 35% to not_complaint.	Hey @UbisoftSupport, I'm still experiencing the jager glitch and it's really affecting my gameplay. Can you please fix it asap? (65% complaint, 35% not_complaint)	complaint
	$\hat{P}(y_i\|x_i)$	0.75			
	miscalibration type	overconfidence			
SUBJ	x_i	the gripping story of the dutch bettien martens , based on true facts.	An example x_i which belongs 95% to objective and 5% to subjective (based on a classifier's categorization). Now I ask you act as that classifier and based on this example, generate a diverse set of 3 short utterances where each utterance belongs 70% to objective and 30% to subjective.	The remarkable tale of Bettien Martens serves as a testament to the enduring nature of hope and determination, even in the face of unimaginable hardship. (70% objective, 30% subjective)	objective
	$\hat{P}(y_i\|x_i)$	0.10			
	miscalibration type	underconfidence			
B77	x_i	Can my teenager have an account?	An example x_i which belongs 65% to age_limit and 35% to atm_support (based on a classifier's categorization). Now I ask you act as that classifier and based on this example, generate a diverse set of 3 short utterances where each utterance belongs 85% age_limit and 15 % to atm_support.	Can my toddler open their own bank account? (85% age_limit and 15% to atm_support)	age_limit
	$\hat{P}(y_i\|x_i)$	0.65			
	miscalibration type	underconfidence			
SE	x_i	The zoom function on this camera is so loud that sometimes you will be unable to use it if you find yourself in a situation where you must be quiet.	An example x_i which belongs 75% to negative and 25% to positive (based on a classifier's categorization). Now I ask you to act as that classifier and based on this example, generate a diverse set of 3 short utterances where each utterance belongs 55% to negative and 45% to positive.	I love how this router can handle a large network, but the price is a bit steep for my taste. (55% negative, 45% positive)	negative
	$\hat{P}(y_i\|x_i)$	0.25			
	miscalibration type	overconfidence			

Table 8: An example of generating synthetic data via LLM. **Input** contains the original text (x_i) and the average predictive probability of the bin it comes from ($\hat{P}(y_i|x_i)$). **Generated Text** is the one after the relabeling process. **Note**: during re-fine tuning of the downstream model, we exclude $\hat{P}(y_i|x_i^{syn})$ and retain only x_i^{syn} in the dataset.