Beyond Confidence: Reliable Models Should Also Consider Atypicality

Mert Yuksekgonul  
Stanford University  
merty@stanford.edu

Linjun Zhang  
Rutgers University  
lz412@stat.rutgers.edu

James Zou†  
Stanford University  
jamesz@stanford.edu

Carlos Ernesto Guestrin†  
Stanford University, CZ Biohub  
guestrin@stanford.edu

Abstract

While most machine learning models can provide confidence in their predictions, confidence is insufficient to understand a prediction’s reliability. For instance, the model may have a low confidence prediction if the input is not well-represented in the training dataset or if the input is inherently ambiguous. In this work, we investigate the relationship between how atypical (rare) a sample or a class is and the reliability of a model’s predictions. We first demonstrate that atypicality is strongly related to miscalibration and accuracy. In particular, we empirically show that predictions for atypical inputs or atypical classes are more overconfident and have lower accuracy. Using these insights, we show incorporating atypicality improves uncertainty quantification and model performance for discriminative neural networks and large language models. In a case study, we show that using atypicality improves the performance of a skin lesion classifier across different skin tone groups without having access to the group attributes. Overall, we propose that models should use not only confidence but also atypicality to improve uncertainty quantification and performance. Our results demonstrate that simple post-hoc atypicality estimators can provide significant value.

1 Introduction

Typicality is an item’s resemblance to other category members [RM75]. For example, while a dove and a sparrow are typical birds, a penguin is an atypical bird. Many works from cognitive science (e.g., [Rip89, RSS73, MP80]) suggest that typicality plays a crucial role in category understanding. For instance, humans have been shown to learn, remember, and refer to typical items faster [Mur04]. Similarly, the representativeness heuristic is the tendency of humans to use the typicality of an event as a basis for decisions [TK74]. This cognitive bias is effective for making swift decisions, but it can lead to poor judgments of uncertainty. For instance, the likelihood of typical events can be overestimated [TK74] or uncertainty judgments can be inferior for atypical events [TK92].

While it is hard to quantify the uncertainty of human judgments, machine learning models provide confidence in their predictions. However, confidence alone can be insufficient to understand the reliability of a prediction. For instance, a low-confidence prediction could arise from an ambiguity that is easily communicated, or due to the sample being underrepresented in the training distribution.

1Our code is available at https://github.com/mertyg/beyond-confidence-atypicality

Atypicality in Uncertainty. Left: We show examples from the ImageNet-R dataset with our atypicality framework. Right: We provide a conceptualization of the quadrants. Using atypicality, we can understand prediction quality (§3), improve predictions (§4), and prediction sets (§5).

Similarly, a high-confidence prediction could be reliable or miscalibrated. Our main proposal is that models should quantify not only the confidence but also the atypicality to understand the reliability of predictions or the coverage of the training distribution. However, many machine learning applications rely on pretrained models that solely provide confidence levels, devoid of any measure of atypicality.

Contributions: To support our position, we use a simple formalization of atypicality estimation. With the following studies, we show that by using simple atypicality estimators, we can:

1. Understand Prediction Quality: Calibration is a measure that assesses the alignment between predicted probabilities of a model and the true likelihoods of outcomes [GR07]. Neural networks [GPSW17] or even logistic regression [BMWX21a] can be miscalibrated out-of-the-box. Here, we argue that using atypicality can give insights into when a model’s confidence is reliable. Through theoretical analysis and extensive experimentation, we demonstrate that atypicality results in lower-quality predictions. Specifically, we show that predictions for atypical inputs and samples from atypical classes are more overconfident and have lower accuracy.

2. Improve Calibration and Accuracy: Recalibration methods offer some mitigation to miscalibration [GPSW17] by adjusting a probabilistic model. We show that models need different adjustments according to the atypicality of inputs and classes, and atypicality is a key factor in recalibration. In light of these findings, we propose a simple method: Atypicality-Aware Recalibration. Our recalibration algorithm takes into account the atypicality of the inputs and classes and is simple to implement. We show that complementing recalibration methods with atypicality improves uncertainty quantification and the accuracy of predictors. Further, in a case study for skin lesion classification, we show that atypicality awareness can improve performance across different skin-tone subgroups without access to group annotations.

3. Improve Prediction sets: An alternative approach to quantify uncertainty is to provide prediction sets that contain the label with high probability [ABMJ20]. Here, we investigate existing methods with atypicality and show that prediction sets could underperform for atypical or low-confidence samples. By using atypicality, we demonstrate the potential for improving prediction sets.

Overall, we propose that models should also consider atypicality, and we show simple- and easy-to-implement atypicality estimators can provide significant value.

2 Interpreting Uncertainty with Atypicality

Motivation: In many machine learning applications, we have access to a model’s confidence, which aims to quantify the likelihood that a prediction will be accurate. In classification, model output is a probability distribution over classes and confidence is the predicted probability of the top class, i.e., \( \max_y \hat{P}(Y = y|X = x) \). In practical scenarios, confidence is the primary tool used to evaluate the reliability of a prediction where higher confidence is associated with better predictions. However, the uncertainty in confidence can stem from different sources that require different treatment [MKvA`21].
Here, we call a prediction reliable if it is high-confidence and well-calibrated. High confidence could be reliable or miscalibrated, and low confidence could be due to ambiguity or rare inputs. We propose that atypicality provides a natural way to understand reliability when combined with confidence. A sample is called typical if it is well-represented in the previously observed samples, e.g., an image of a dog that is similar to other dogs in the training data. However, if the image is unlike any other seen during training, it is atypical. We argue that atypicality can help us interpret a prediction’s reliability. Below we categorize samples and predictions according to atypicality and confidence in four quadrants (Figure 1).

**High-confidence and representative:** Reliable predictions often fall within the Reliable Quadrant, which includes typical, high-confidence samples. These samples are well-represented in the training dataset (typical), thus we expect the high-confidence prediction to be reliable. For instance, the first image on the top left (Figure 1) is a typical golden retriever and the model makes a reliable prediction.

**High-confidence yet far from the support:** Having high-confidence does not always indicate reliability. If the sample does not have support in the training distribution, the confidence could be miscalibrated. Such samples lie in the Extrapolation Quadrant which contains atypical, high-confidence samples. For instance, the second image in the top right of Figure 1 is a toy hog and the model has not seen similar ones during training.

**Low confidence due to ambiguity:** In contrast, low confidence could also be reliable when it correctly reflects an ambiguity. Such samples are in the Ambiguous Quadrant that contains typical, low-confidence samples. These are typical since they may represent multiple classes; yet, due to ambiguity, the model’s confidence is low. For instance, the second image in the bottom left of Figure 1 can both be a hog and a comic book.

**Low confidence and rare:** For samples that are not well-represented in training data, we expect to have low-quality predictions. Untrustworthy Quadrant comprises atypical, low-confidence samples that can include extremely rare subgroups, for which we expect miscalibration and lower accuracy. For example, the image in Figure 1 bottom right is an origami hog that was not seen in training.

These examples suggest that relying solely on confidence does not provide a complete understanding of the reliability of the predictions, and we can use atypicality to interpret and improve reliability.

**Formalizing Atypicality:** Atypicality here is defined with respect to the training distribution. Informally, an input or a class is atypical if it is not well-represented in the training distribution. For instance, if there are no or limited similar examples to an input, it can be called atypical. Note that this notion is not restricted to being ‘out-of-distribution’ [HG16a], since in-distribution groups could also be atypical or rare, and our goal is to perform reliably for the entire spectrum.

Formally, let $X \in \mathbb{R}^d$ be the random variable denoting features and $Y \in \mathcal{Y} = \{1, 2, ..., C\}$ denote the class, where we focus on classification.

**Definition 2.1 (Input Atypicality).** We define the atypicality of the input $x$ as$^2$

$$a_X(x) = - \max_y \log \mathbb{P}(X = x | Y = y).$$

We use the logarithm of the class-conditional densities due to high dimensionality and density values being close to zero. Intuitively, for a dog image $x$, if $\mathbb{P}(X = x | Y = \text{dog})$ has a low value, we call $x$ an atypical dog image. Overall, if $a(x)$ is high, then we call $x$ an atypical input. Specifically, if an input is not typical for any class, then it is atypical with respect to the training distribution. Similarly, we can also use marginal density, $\mathbb{P}(X = x)$, or distance$^3$ to quantify atypicality.

Similarly, the notion of atypical (rare) classes is prevalent in imbalanced classification [CWG+19, ZCLJ21]. Ensuring reliable performance for atypical classes can be safety-critical, e.g., for a rare presence of dangerous melanoma [DVN+22]. We define class atypicality in the following:

**Definition 2.2 (Class Atypicality).** For a class $y$, atypicality of a class is defined as

$$a_Y(y) = - \log \mathbb{P}(Y = y).$$

$^2$Here atypicality differs from ‘typical sets’ in information theory that refers to a sequence of variables [TJ06].

$^3$For an input $x$, if the nearest neighbor (NN) distance is large, then we call $x$ atypical as all inputs in the training set are far from $x$. Density and distance are connected through non-parametric density estimation and [JKGG18] shows that NN distance can recover high-density regions.

$^4$When the meaning is unambiguous, we omit the subscript to denote $a(X)$ or $a(Y)$ for notational brevity.
Estimating Atypicality for Discriminative Models: Quantifying input atypicality requires access to the class-conditional / marginal distributions. In practice, for neural networks trained for classification, these distributions are unavailable and we need to perform the estimation. This estimation can be challenging if the dimensionality is large, or the data is unstructured, requiring assumptions about the distributions. Prior works [MKvA+21, LLLS18] showed that Gaussian Mixture Models (GMMs) in the embedding space of neural networks can be used to model these distributions.

In experiments, we use Gaussians with shared covariance, i.e. \( \hat{p}(X = x \mid Y = c) \sim N(\mu_c, \Sigma) \), to estimate input atypicality. We perform the estimation in the penultimate layer of neural networks used to make predictions, using maximum-likelihood estimation with samples from the training data. We explore other metrics, such as \( k \)-Nearest Neighbors distance. We give implementation details and results with different metrics in Appendix A.1. With these estimators, atypicality estimation is cheap and can run on a CPU. Our goal is to show that simple estimators can already reap large benefits. Our framework is flexible and exploring more sophisticated estimators is a topic for future work.

Atypicality for LLMs: LLMs are increasingly used for classification [BMR+20]. Modern LLMs are autoregressive models that compute a marginal distribution, \( \hat{p}_{\text{LLM}}(X) \). We compute the negative log-likelihood of a prompt or a label and use this as an atypicality metric, i.e. \( a_X(x) = -\log \hat{p}_{\text{LLM}}(x) \), \( a_Y(y) = -\log \hat{p}_{\text{LLM}}(y) \). Similar to the discriminative setting, atypicality here aims to quantify whether a prompt is well-represented in the training data. A larger value for \( a(X) \) would imply that a prompt is more atypical. Below, we present typical and atypical prompts for the AGNews dataset:

Classify the news articles into the categories of World, Sports, Business, and Technology.

Article: Sinfin tallest obstacle to host #39;s patriotic games hope AS tennis fans go, Houston #39;s Jim #39;Mattress Mack #39; McIngvale is very rich, extremely forthright, exceedingly patriotic and unflinchingly Republican.
Answer: Atypicality: 353.45, Percentile: %94.5

Article: Delta Air Lines Prepares Chapter 11 Filing Delta Air Lines Inc. could file for Chapter 11 bankruptcy protection as soon as next week, a source familiar with the matter said yesterday.
Answer: Atypicality: 171.50, Percentile: %0.9

3 Understanding the Prediction Quality with Atypicality

In this section, we show how our framework can be applied to understand the quality of predictions.

Experimental Setup: We investigate three classification settings across a range of datasets:

1. **Balanced Supervised Classification:** We use ResNet18-50-152 [HZRS16], WideResNet28 [ZK16], RoBERTa [LOG+19] trained on ImageNet [DDS+09], CIFAR10,100 [Kri09], MNLI [WNB18] respectively.
2. **Imbalanced Supervised Classification:** We use ResNet18, ResNext50, ResNet152 trained on CIFAR-LT, ImageNet-LT and Places365-LT where models and data are mostly from [ZCLJ21, MKS+20]. Note that all of the test and validation sets have balanced class distributions.
3. **Classification with LLMs:** We use open-source Alpaca7B [TGZ+23] on IMDB [MDP+11], TREC [LR02], and AG News [ZZL15] datasets with the prompts from [ZWF+21].

Details on datasets, models, and prompts are in Appendix B. Our experiments were run on a single NVIDIA A100-80GB GPU. We report error bars over 10 random calibration/test splits.

3.1 Atypicality is Correlated with Miscalibration

We first explore the importance of atypicality to understand model calibration. Calibration quantifies the quality of a probabilistic model [GR07]. Informally, a model is considered perfectly calibrated if

\[
\hat{P}(Y \mid \hat{P}(X) = p) = p. 
\]

For the sake of simplicity, consider a binary classification problem where the predictor is \( \hat{P} : \mathcal{X} \to [0, 1] \). We quantify miscalibration with Calibration Error (CE):

\[
CE(\hat{P}) = \mathbb{E}[|\hat{P}(Y \mid \hat{P}(X) = p) - p|].
\]

It is computationally infeasible to calculate the above expectation with the conditional probability \( \hat{P}(Y \mid \hat{P}(X) = p) \). In practice, we use a binned version of this quantity. Expected Calibration Error (ECE) [NCH15, GPSW17], to estimate CE. See Appendix C.1 for a formal definition.
Here, we aim to examine the relationship between model calibration and atypicality. Given any $K > 1$, we consider the quantiles of $a(X)$, $a_1, a_2, \ldots, a_{K+1}$ such that $P(a(X) \in (a_k, a_{k+1}]) = 1/K$ for $k \in [K]$. For imbalanced classification problems, we compute the quantiles using the class atypicality. Specifically, we investigate the atypicality-conditional calibration error $ECE[\hat{P} | a(X) \in (a_k, a_{k+1}])$, i.e., the expected calibration error of an input that falls within the atypicality quantile $k$.

### Atypical Examples are Poorly Calibrated
In Figure 2a, we show the distribution of miscalibration where each bin within the grid contains the intersection of the corresponding confidence and atypicality quantiles. We observe that within the same confidence range, predictions for atypical points have lower accuracies and are more overconfident. In other words, predictions in the Extrapolation or Untrustworthy regions are more miscalibrated than the ones in the typical regions.

In Figure 2b, we split inputs into quantiles according to atypicality and compute the ECE and Accuracy for each group. Results show a monotonic relationship between atypicality and ECE or Accuracy across the three settings. Specifically, we see that predictions for atypical inputs or samples from rare classes are more miscalibrated and have lower accuracy. For samples from rare classes, the model overpredicts the probabilities of the typical class, hence we have overconfidence and low accuracy. Appendix C.3, and §4 present figures and tables for all model and dataset pairs.

#### 3.2 Theoretical Analysis: Characterizing Calibration Error with Atypicality

We characterize how calibration error varies with atypicality in a tractable model that is commonly used in machine learning theory [BMWZX21a, BMWZX21b, ZDKZ22, CLKZ22]. Our theoretical analysis further supports our empirical findings.

### Data Generative Model
We consider the well-specified logistic model for binary classification with Gaussian data, where $Y \in \{-1, 1\}$ and the $P(Y = 1|X)$ is defined by the sigmoid function:

\[\mathbb{P}(Y = 1 \mid X) = \sigma(\beta^\top X), \quad X \sim N(0, I_d).\]

Where $I_d$ denotes the $d$-dimensional identity matrix, $\beta^*$ is the ground truth coefficient vector, $\sigma(x) = 1/(1 + e^{-x})$, and we have i.i.d. observations \{(\xi_i, y_i)\}_{i=1}^n sampled from the above distribution.

### The Estimator
We focus on studying the solution produced by minimizing the logistic loss

\[\hat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(\beta^\top x_i)) - y_i \cdot \beta^\top x_i.\]

For $k \in \{-1, 1\}$, $\hat{\mathbb{P}}_k(x)$ is an estimator of $\mathbb{P}(y = k|x)$, with the form $\hat{\mathbb{P}}_k(x) = \frac{1}{1 + e^{-\beta^\top x_i}}$.

**Calibration:** We consider all $x$ where $\mathbb{P}_1(x) > 1/2$, as $\mathbb{P}_1(x) \leq 1/2$ can be analyzed similarly by symmetry (see Appendix G). For $u \in (1/2, 1)$, the signed calibration error at a confidence level $u$ is

\[u - \mathbb{P}(Y = 1 \mid \hat{\mathbb{P}}_1(X) = u).\]
We want to show that when $X$ is atypical, i.e., when $a(X) := \|X\|^2/2$ is larger\(^5\), the accuracy $\mathbb{P}(Y = 1 \mid \hat{P}_1(X) = u)$ would be generally smaller than the confidence $u$ (over-confidence).

**Theorem 3.1.** Consider the data generative model and the learning setting above. For any $K > 1$, suppose we consider the quantiles of $a(X)$, $a_1, a_2, \ldots, a_K, a_{K+1}$ such that $\mathbb{P}(a(X) \in (a_k, a_{k+1}]) = 1/K$ for $k \in [K]$. We assume $\|\beta^*\| \leq c_0$, and $d/n = \kappa$, for some sufficiently small $c_0$. Then, for sufficiently large $n$, for $k = 2, \ldots, K$, we have

\[
\begin{align*}
E_{u \sim \hat{P}_1(X)}[u - \mathbb{P}(Y = 1 \mid \hat{P}_1(X) = u) \mid a(X) \in (a_k, a_{k+1}] > \\
E_{u \sim \hat{P}_1(X)}[u - \mathbb{P}(Y = 1 \mid \hat{P}_1(X) = u) \mid a(X) \in (a_{k-1}, a_k]] \geq 0.
\end{align*}
\]

That is, the resulting classifier is over-confident, and the level of over-confidence becomes larger when the data is more atypical (with larger $a(X)$). Further, the gap becomes larger for smaller sample sizes $n$. The proof of the theorem is in Appendix G.2 and builds on the results from [BMWX21a, SC19].

### 4 Using Atypicality to Improve Recalibration

Here, we show how atypicality can complement and improve post-hoc calibration. In §2, we observed that predictions for atypical inputs and samples from atypical classes are more overconfident with lower accuracy. We next show that taking input and class atypicality into account improves calibration.

#### 4.1 Parametric Recalibration: Different Groups Need Different Temperatures

Temperature scaling (TS), a single parameter variant of Platt Scaling [P+99], is a simple recalibration method that calibrates the model using a single parameter. The predictor is of the form

$$
\log \hat{P}_{TS}(Y\mid X) \propto \log \hat{P}(Y\mid X)/\tau,
$$

where $\hat{P}(Y\mid X)$ is the model that takes an input and outputs scores/logits, and $\tau$ is the temperature parameter. In practice, $\tau$ is optimized using a calibration set to minimize a proper scoring rule [GR07, BW19] such as the cross-entropy loss.

To understand the behavior of TS with respect to atypicality, we separately perform TS on points grouped according to the atypicality quantiles. Let us denote the temperature fitted to the quantile covering $a(X) \in (a_{k-1}, a_k]$ by $\tau_{a_k}$. In Appendix Figure 10 we observe an increasing relationship between $a_k$ and $\tau_{a_k}$. Different atypicality groups need different adjustments, and more atypical groups need larger temperatures. This suggests that being atypicality-aware can improve calibration. While a single temperature value improves average calibration, it may hurt certain groups.

#### 4.2 Atypicality-Aware Recalibration

We showed that predictions are more reliable when the input is typical. However, predictions are less reliable for atypical inputs, and we may need further revision. An analogy can be drawn to decision-making literature where opinions of individuals are combined with geometric averaging weighted by their expertise [FP98, AR89]. Analogously, we propose Atypicality-Aware Recalibration (AAR) a method designed to address the reliability issues identified in dealing with atypical inputs:

$$
\hat{P}_{AAR}(Y\mid X) = \frac{\hat{P}(Y\mid X)^{\psi(a(X))} \exp(S_Y^{1-\psi(a(X))})}{Z(X)},
$$

where $\psi(a(X))$ is a function of input atypicality, $S_Y$ is a tunable score for class $Y$, $Z(X)$ is the normalization term. Intuitively, when the input is typical, we trust the model confidence; otherwise, we use a score for the given class estimated from the calibration set. Note that this form simplifies to

$$
\log \hat{P}_{AAR}(Y\mid X) \propto \phi(a(X)) \log \hat{P}(Y\mid X) + S_Y,
$$

\(^5\)The definition of atypicality follows from the marginal likelihood of the data model: density for the Gaussian with zero mean and identity covariance.
where we subsume \(1 - \psi(a(X))\) into \(\phi(a(X))\). We give a simple interpretation of this form: the multiplicative term is an atypicality-dependent temperature, and the additive term is a class-dependent correction where \(\exp(S_Y)\) can be considered to induce a correction distribution over classes estimated from the calibration set.

Intuitively, when \(\psi(a(X)) = 0\), the output reduces to a fixed distribution over classes that was estimated using the calibration set. This distribution can be seen to induce a prior probability over classes, and \(\psi\) controls the tradeoff between this prior and the model’s predictive distribution. As the point becomes more typical, this distribution is closer to the model’s predictive distribution. In Appendix Figure 11, we show how these values behave with class atypicality. We find that rare classes require larger positive corrections with larger \(S_Y\).

**Implementation Details:** Following TS, we minimize the cross-entropy loss on a calibration set. With the temperature-atypicality relationship observed in Figure 10 we choose to instantiate the multiplicative factor as a quadratic function, where \(\phi(a(X)) = c_2a(X)^2 + c_1a(X) + c_0\) and in total we have \(|\{S_1, ..., S_|Y|, c_0, c_1, c_2\}| = |Y| + 3\) interpretable parameters. Once the embeddings and logits are computed, AAR runs on a CPU in under 1 minute for all experimental settings.

Similar to our adaptive interpretation, a concurrent work, Adaptive Temperature Scaling (AdaTS) [JPL+23], uses temperature scaling where the temperature is parameterized by a Variational Autoencoder (VAE) [KW15] and a multi-layer perceptron on top of the VAE embeddings. In the below experiments, we give results with AdaTS as a baseline when applicable.
When the goal is to improve performance across groups, one can use group annotations and optimize our findings suggest that Atypicality-Awareness can complement fairness-enforcing methods, and improve performance even when the group annotations are unavailable. We hypothesize that with AAR, we can perform better than using supervised group attributes since groups may not have sufficient sample size in the calibration set (131, 1950, 1509, 555 samples for Unknown, 1&2, 3&4, and 5&6 respectively), and we can leverage atypicality to offer some mitigation. Further investigating how to leverage atypicality to improve fairness and factors affecting performance disparities is a promising direction for future work [CJS18].

4.3 Case Study: Fairness through Atypicality-Awareness

Machine learning models reportedly have performance disparity across subgroups [BHN17] due to factors such as varying sample size or noise levels [CJS18]. For instance, skin lesion classifiers can exhibit performance disparity across different skin tones [DVN+22]. Fitzpatrick17k [GHS+21] is a dataset of clinical images with Fitzpatrick skin tone annotations between 1-to-6, where a larger number means darker skin tones, and when annotators do not agree, it is labeled as ‘Unknown’. We explore the classification problem with 9 classes indicating the malignancy and the type of skin condition, using a ResNet18/34 pretrained on ImageNet and finetuned on this task (See Appendix F).

When the goal is to improve performance across groups, one can use group annotations and optimize performance within each group [HJKRR18, KGZ19]. Here, we investigate how complementing recalibration with atypicality can improve prediction quality across all groups without group annotations. For comparison, we perform 3 recalibration methods: TS, AAR, and Skin-Tone Conditional. For example, we perform 3 recalibration methods: TS, AAR, and Skin-Tone Conditional. TS which calibrates the model individually for each skin-tone group with TS. Since the skin-tone conditional calibration uses group attributes, ideally it should act as an oracle.

In Figure 4, we give the Accuracy and ECE analyses where AAR improves performance across all groups. For instance, the worst-group Accuracy (0.69) or ECE (0.072) with AAR is close to the best-group Accuracy (0.63) or ECE (0.062) with the other two methods. Overall, our findings suggest that Atypicality-Awareness can complement fairness-enforcing methods, and improve performance even when the group annotations are unavailable. We hypothesize that with AAR, we can perform better than using supervised group attributes since groups may not have sufficient sample size in the calibration set (131, 1950, 1509, 555 samples for Unknown, 1&2, 3&4, and 5&6 respectively), and we can leverage atypicality to offer some mitigation. Further investigating how to leverage atypicality to improve fairness and factors affecting performance disparities is a promising direction for future work [CJS18].

5 Improving Prediction Sets with Atypicality

Conformal Prediction [SV08, AB21] is a framework that assigns a calibrated prediction set to each instance. The goal is to find a function $C : \mathcal{X} \rightarrow 2^Y$ that returns a subset of the label space such that $Y \in C(X)$ with high probability. The framework aims to guarantee marginal coverage, i.e., $P(Y \in C(X)) \geq 1 - \alpha$, for a choice of $\alpha$. We investigate two conformal calibration methods,
Adaptive Prediction Sets (APS) [RSC20] and Regularized APS (RAPS) [ABMJ20]. Let \( \pi(X) \) be the permutation of the label set that sorts \( \hat{P}(Y = c|X) \), i.e. the predicted probabilities for each class \( c \) after TS. The prediction sets are produced by the function \( \hat{C}(x) = \{ y : s(x, y) \leq \hat{q} \} \), and these methods fit the threshold \( \hat{q} \) for a choice of the scoring function. APS uses the cumulative sum of the predicted probabilities \( s(x, y) = \sum_{j=1}^{c} \hat{P}(Y = j|X) \), where \( y = \pi_c(X) \). Intuitively, if the model was perfectly calibrated, we would have expected to have \( \hat{q} = 1 - \alpha \). Similarly, RAPS builds on the idea that tail probabilities are noisy and regularizes the number of samples in the prediction set.

Building on our ideas in the previous sections we implement Atypicality-Aware prediction sets, namely AA-APS and AA-RAPS in the following way: We first group points according to their confidence and atypicality quantiles. A group \( G \) here is defined using 4 thresholds, namely \( G = x : (l_{a}^{(G)} < q_{a}(x) \leq h_{a}^{(G)}) \land (l_{c}^{(G)} < q_{c}(x) \leq h_{c}^{(G)}) \) where \( q_{a}(x) \) and \( q_{c}(x) \) denote the atypicality and confidence quantiles for the sample \( x \), \( l_{a}^{(G)} \) and \( h_{a}^{(G)} \) denote the atypicality lower and upper bounds for group \( G \), and \( l_{c}^{(G)} \) and \( h_{c}^{(G)} \) denote the confidence lower and upper bounds for group \( G \). Using a calibration set, these bounds are simply determined by the quantiles of confidence and atypicality statistics. Then, we fit separate thresholds \( \hat{q}_{G} \) for each group’s prediction sets with APS or RAPS as subroutines. This allows us to have an adaptive threshold depending only on the atypicality and confidence of predictions.

![Figure 5: Improving Conformal Calibration with Atypicality for ResNet50 on ImageNet. Here we show that atypicality awareness improves conformal calibration performance across different groups. Methods are fitted to satisfy 95% coverage. We observe that APS and RAPS do not satisfy conditional coverage for high atypicality regions or low confidence regions.](image)

In Figure 5, we provide the coverage plots for APS and RAPS in the first and third columns. Even though marginal coverage is satisfied, models do not satisfy conditional coverage for atypical inputs or low-confidence predictions. We observe that being Atypicality-Aware improves coverage across otherwise underperforming groups. Further, AA-APS has lower set sizes on average than APS (15.6 vs 21.3). While RAPS has a lower average set size than AA-RAPS (4.2 vs 9.1) AA-RAPS has smaller set sizes for high-confidence samples, whereas a larger set size for low-confidence samples where the coverage is not met for RAPS. In Appendix D.3, we provide the same analysis for ResNet18,50,152 at different coverage levels along with analyzing the performance in the Confidence and Atypicality dimensions individually. For instance in Figure 8, we observe that RAPS and APS do not satisfy coverage for high atypicality regions, even when averaged across different confidence levels.

### 6 Additional Related Work

**Uncertainty and Atypicality:** [MKvA+21, PBC+20] use density estimation to disentangle epistemic and aleatoric uncertainty. Following this, they show improvements in active learning and OOD detection [LLLS18]. We note that our goal is not this disentanglement (e.g. Untrustworthy quadrant can have both aleatoric or epistemic uncertainty), or Ambiguity could be due to a lack of features or noise. [LLP+20] propose the related notion of distance awareness, and that it leads to better uncertainty quantification. They offer architecture and training modifications whereas we analyze existing models using our framework including imbalanced and LLM settings, and propose simple and post-hoc approaches. ‘OOD’ [HG16b] or ‘anomaly’ [HMD18] notions are tied to atypicality, yet our goal is not to make a binary distinction between ‘in’ or ‘out’. We argue that in-distribution samples could also be atypical (e.g. rare groups), and the goal is to perform reliably in the entire spectrum. Other works with an atypicality notion include bounding calibration of groups by the
excess risk [LSH19], miscalibration under distribution shifts [OFR+19], uncertainty in Gaussian Processes [Ras04], forgetting time for rare examples [MGLK22], the poor performance of groups with lower sample sizes [CJS18], energy-based models improving calibration [GWJ+20], relating perplexity to zero-shot classification performance for LLMs [GIB+22], grouping loss and local definitions of miscalibration [PLMV23], the relationship between active learning and atypicality [HDW22], sample size as a factor for subgroup performance disparity [CJS18]. [PBZ21] provide insightful discussion around the nature of softmax confidence, and here we show that its reliability depends on the atypicality of the input. Our new findings include showing that predictions for atypical samples are more miscalibrated and overconfident, and atypicality awareness improves prediction quality. Overall, while there are other relevant notions in the literature, our distinct goal is to show that post-hoc atypicality estimation and recalibration is a simple yet useful framework to understand and improve uncertainty quantification that complements existing methods.

Recalibration and Conformal Prediction: There is a rich literature on recalibration methods and prediction sets: TS [GPSW17], Platt Scaling [P+99], conformal calibration [SV08, ABMJ20] among many. [LLC+22, RBSC19, BYR+21, BGJ+22] make a relevant observation, showing that the coverage of conformal prediction is not equal across all groups. They propose group conformal calibration, which requires group labels whereas our proposal is unsupervised and does not depend on any attribute information. Concurrent work [JPL+23] explores AdaTS, where they train a separate VAE and MLP to produce an adaptive temperature. However, our parameterization of temperature has 3 parameters and is interpretable.

7 Conclusion

Atypicality offers a simple yet flexible framework to better understand and improve model reliability and uncertainty. We propose that pretrained models should be released not only with confidence but also with an atypicality estimator. While there are other relevant notions in the literature, our main goal is to show that atypicality can provide a unifying perspective to discuss uncertainty, understand individual data points, and improve fairness. Here we focus on classification problems; it would be interesting to extend atypicality to regression and generation settings. Furthermore, we would like to extend the theoretical analysis to more general settings, as our empirical results demonstrate that the observed phenomena hold more broadly.

Acknowledgments

We would like to thank Adarsh Jeewajee, Bryan He, Edward Chen, Federico Bianchi, Kyle Swanson, Natalie Dullerud, Ransalu Senanayake, Sabri Eyuboglu, Shirley Wu, Weixin Liang, Xuechen Li, Yongchan Kwon, Yu Sun, and Zach Izzo for their comments and suggestions on the manuscript. Linjun Zhang’s research is partially supported by NSF DMS-2015378. Carlos Ernesto Guestrin is a Chan Zuckerberg Biohub – San Francisco Investigator.

References


13


A Atypicality Estimation

A.1 Input Atypicality Estimation

To estimate input atypicality, we use two ways to estimate the likelihood of a point under the training distribution. First, we give methods for the discriminative models.

Fitting individual Gaussians to class conditionals Here, we follow a similar approach to [MKvA + 21]. Namely, we model the class conditionals with a Gaussian distribution, where the covariance matrix is tied across classes:

\[ \hat{P}(X|Y = y) \sim N(X; \mu_y, \Sigma) \]  

We fit the parameters \( \mu_y \) and \( \Sigma \) with maximum likelihood estimation. The reason to tie the covariance matrix is due to the number of samples required to fit the density. Namely, for a \( d \)-dimensional problem, the total number of parameters to fit individual matrices becomes \( O(yd^2) \), which results in low-quality estimates. Then, the atypicality becomes

\[ a_X(x) = -\max_{y \in Y} \log \hat{P}(X = x|Y = y) \]  

Computing distance with k-Nearest Neighbors k-Nearest Neighbors: Similarly, we can use the nearest neighbor distance. Concretely, we use the nearest neighbor distance, \( a_X(x) = d_{\min}(x, D_{\text{train}}) = \min_{x' \in D_{\text{train}}} |x' - x| \), as the atypicality metric. Alternatively, we can use different notions such as the average of k-nearest neighbors, or the distance to the kth neighbor. Below, we report the results by using the average distance to 5-nearest neighbors.

Fitting the estimators For all of the atypicality estimators, we fit the estimators using samples from the training sets and make inferences for the calibration and test sets. For instance, we use the training split of ImageNet to fit the corresponding density estimator and compute the atypicality for the samples from the validation/test split of ImageNet. All of our results using atypicality are reported for the test splits of the below datasets.

Atypicality Estimation with LLMs: For language models we simply compute the negative log-likelihood of each prompt as the atypicality metric:

\[ a_X(x) = -\log \hat{P}_{\text{LLM}}(x) \]  

To define confidence, we use the logits of the language model, conditioned on the prompt. We use the logit of the first token of each class label and compute the predicted probabilities by applying softmax to the logits of each class.

A.2 Class Atypicality

To estimate class atypicality, we simply count the fraction of examples from a particular label in the training dataset. Let us have a training dataset \( D_{\text{train}} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \). Then, we estimate class atypicality with

\[ a_Y(y) = -\log \frac{\sum_{i \in [N]} 1[y_i = y]}{N} \]  

A.3 Atypicality and Confidence

Are atypicality and confidence equally informative? Beyond the data perspective given in Figure 1, here we provide quantitative results to demonstrate the difference. In Figure 6, we give a grid plot where the x-axis indicates the typicality quantile of a point, and the y-axis indicates the confidence of a point. The coloring on the left indicates the accuracy within a bin split according to accuracy; and the right has the difference between average confidence and accuracy. Observe that for a specific confidence interval, larger values of typicality mean better quality probabilistic estimates and larger atypicality means more miscalibration.
Figure 6: Input Atypicality and Confidence. Here, the x-axis reflects the input atypicality quantile, and the y-axis indicates confidence. The coloring for the figure on the left indicates the accuracy within a bin, and the figure on the right has the difference between confidence and accuracy within a bin. We observe that even within the same confidence range, atypical examples tend to be more miscalibrated and overconfident compared to typical examples.

B Experimental Details

B.1 Balanced Supervised Classification

B.1.1 Datasets

Below is a full list of datasets for balanced classification:

1. **ImageNet** [DDS+09] from Torchvision [MR10] is an object recognition dataset with 1000 classes. We use the ImageNet-1k version.

2. **CIFAR10/100** [Kri09] from Torchvision [MR10] are object recognition datasets with 10/100 classes.

3. **MNLI** [WNB18] from Huggingface Datasets [LVdMJ+21] is a natural language inference dataset with 3 classes, indicating entailment, neutral, and contradiction outcomes.

B.1.2 Models

Most of the models are public models, e.g., obtained from the Transformers Library [WDS+20] or Torchvision [MR10]. Below we give the full model details and how one can access them:

1. **RoBERTa** (HuggingFace roberta-large-mnli) trained on the MNLI dataset. One can use the id given here on HuggingFace to download the model.


3. **WideResNet28** trained on CIFAR10,100 obtained from [MKS+20].

For all BERT [DCLT19] style models we use the [CLS] token embeddings in the final layer to perform classification. For all vision models, we use the penultimate layer embeddings to fit the density estimators and perform the analyses. In the experiments, we randomly split the test sets into two equal halves to have a calibration split and a test split, and repeat the experiments over 10 random seeds.
B.2 Imbalanced Supervised Classification

B.2.1 Datasets

All of our imbalanced classification datasets are previous benchmarks obtained from the GitHub repository'* of [ZCLJ21] with corresponding training, validation, and test splits. All of these datasets have an exponential class imbalance.

1. **ImageNet-LT** is the long-tailed variant of ImageNet with 1000 classes.
2. **CIFAR10/100-LT** is the long-tailed variant of CIFAR10/100 with 10/100 classes.
3. **Places365-LT** is the long-tailed variant of Places365 [ZKL*16] with 365 classes.

B.2.2 Models

Similarly, most of these models are obtained from [ZCLJ21].

1. **ResNeXt50** trained on ImageNet-LT with and without Progressive Balancing, which is a strategy to address class imbalance during training.
2. **ResNet152** trained on Places365-LT
3. **ResNet18** trained on CIFAR100-LT trained by us. This model is pretrained on ImageNet and finetuned on CIFAR100-LT.

We use the validation splits of these datasets as the calibration set, and report the results on the test set.

B.3 Classification with LLMs

B.3.1 Model

We use Alpaca-7B [TGZ*23] in a zero-shot setting, where we simply prompt the model with the classification question. We use the prompting strategy from Content-Free Calibration [ZWF*21].

Below, we show examples of each dataset and prompt.

B.3.2 Datasets

**IMDB** is a binary classification dataset of movie reviews, where the goal is to classify the sentiment in a review. The example prompt has the form ‘The following review was written for a movie: [Review]. What is the sentiment, Positive or Negative? Answer: ’. Below is an example:

The following review was written for a movie: I and a friend rented this movie. We both found the movie soundtrack and production techniques to be lagging. The movie’s plot appeared to drag on throughout with little surprise in the ending. We both agreed that the movie could have been compressed into roughly an hour giving it more suspense and moving plot.

What is the sentiment, Positive or Negative? Answer:

where the correct answer should be ‘Negative’. We filter out the examples that exceed the context length limit of Alpaca7B (512). We noticed that the ‘validation’ split of IMDB leads to significantly worse calibration compared to splitting the test set. Thus, for all experiments, we use the test split of IMDB and split it into 2 sets (instead of using the validation split as a calibration set as in the other two datasets).

**TREC** is a 6-class question classification dataset where the goal is to predict whether a question will have an answer that is an ‘Abbreviation’, ‘Entity’, ‘Description’, ‘Human’, ‘Location’, or a ‘Number’. We format the prompts with ‘Classify the questions based on their Answer Type. Potential Answer Types are: Number, Location, Person, Description, Entity, or Abbreviation. Question: [question] Answer Type: ’. Below is an example prompt:

---

*https://github.com/dvlab-research/MiSLAS
Classify the questions based on their Answer Type. Potential Answer Types are: Number, Location, Person, Description, Entity, or Abbreviation.

Question: What county is Modesto, California in? Answer Type: Location

where the correct answer should be ‘Location’.

AG News is a news classification dataset. The goal is to classify a given news into 4 potential classes: ‘World’, ‘Sports’, ‘Business’, or ‘Science and Technology’. We format the prompts with ‘Classify the news articles into the categories of World, Sports, Business, and Technology.’

Article: Wall St. Bears Claw Back Into the Black (Reuters) Reuters - Short-sellers, Wall Street’s dwindling band of ultra-cynics, are seeing green again.

Answer: Business

where the correct answer should be ‘Business’.

Furthermore, we also use [ZWF+21] as another calibration baseline. Following their paper, we use N/A, [MASK], and the empty string as content-free. Concretely, we follow their paper to first obtain the average predicted probabilities for each label token for the content-free input, denoted by $p_{cf}$. We then let $W = \text{diag}(p_{cf})^{-1}$.

When making test-time predictions, we compute $\text{Softmax}(W^T p)$ as the new predicted probabilities. In our experiments, we observe that it does not perform as well in this setting, as was previously suggested by [Kum22].

C Calibration

We run all our experiments with 10 different random seeds, where the seeds are $\{0, 1, 2, \ldots, 9\}$. Randomness is over fitting the atypicality estimators, and calibration-test splits (we use the same splits with the recalibration experiments for the sake of consistency).

C.1 Expected Calibration Error

To compute ECE, we generate $B = \{B_1, B_2, \ldots, B_M\}$, $M$ equally-spaced bins where samples are sorted and grouped according to their confidence. $B_m$ here denotes the set of the data indices where the confidence of the prediction for the sample falls into the interval $(\frac{m-1}{M}, \frac{m}{M}]$. We compute ECE by

$$ECE[\hat{p}] = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)|,$$  

(7)

where $\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i=1}^{B_m} \mathbf{1}[\hat{y}_i = y_i]$ is the accuracy for the bin $m$, and $\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i=1}^{B_m} \hat{p}(Y = \hat{y}_i|X = x_i)$ gives the average confidence within the bin. $|B_m|$ is the size of the bin $m$, $N$ is the total number of samples, and $\mathbf{1}[\cdot]$ is the indicator function.

Throughout our experiments, we let the number of bins $|B| = 10$ by default when computing ECE.

Similarly, below we report results with RMSCE (Root Mean Squared Error) [HMD18] as another calibration metric, which is formulated as the following:

$$\text{RMSCE}[\hat{p}] = \sqrt{\sum_{m=1}^{M} \frac{|B_m|}{N} (\text{acc}(B_m) - \text{conf}(B_m))^2}$$  

(8)
C.2 ECE and Atypicality Results with Different Atypicality Metrics

We further experiment with different atypicality metrics, such as the average distance to the 5-nearest neighbors (Figure 7). We broadly observe that while there are slight differences in the quantitative results between different atypicality metrics, the qualitative phenomena remain intact. In Tables 1, 2, and 3 we give all the results in the tabular form.

Figure 7: Atypicality with 5-nearest neighbors and Uncertainty. Here, we report the results of the same experiments as Figure 3 with the average of the distance to the 10-nearest neighbors as the atypicality metric. See Tables 1 for the results in tabular format.

C.3 Results in the Tabular Format
Table 1: Recalibration Results for Balanced Classification. For each dataset and atypicality quantile, the best results are marked in bold. We provide the standard errors next to the means over 10 random seeds.

<table>
<thead>
<tr>
<th>Atypicality</th>
<th>Uncalibrated</th>
<th>AdaTS</th>
<th>Atypicality-Aware</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ECE</td>
<td>RMSE</td>
<td>ECE</td>
</tr>
<tr>
<td><strong>GNN</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResNet152</td>
<td>0.2</td>
<td>0.05</td>
<td>0.01</td>
</tr>
<tr>
<td>ImageNet</td>
<td>0.4</td>
<td>0.09</td>
<td>0.01</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>0.6</td>
<td>0.10</td>
<td>0.01</td>
</tr>
<tr>
<td>MNLI</td>
<td>0.8</td>
<td>0.12</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>RoNet18</strong></td>
<td>1.0</td>
<td>0.14</td>
<td>0.01</td>
</tr>
<tr>
<td>ImageNet</td>
<td>0.2</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>0.4</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>MNLI</td>
<td>0.6</td>
<td>0.06</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>ResNet50</strong></td>
<td>0.8</td>
<td>0.08</td>
<td>0.01</td>
</tr>
<tr>
<td>ImageNet</td>
<td>0.2</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>0.4</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>MNLI</td>
<td>0.6</td>
<td>0.06</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>RoBERTs</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MNLJ</td>
<td>1.0</td>
<td>0.07</td>
<td>0.01</td>
</tr>
<tr>
<td>CNN</td>
<td>0.2</td>
<td>0.03</td>
<td>0.01</td>
</tr>
<tr>
<td>ResNet28</td>
<td>0.4</td>
<td>0.06</td>
<td>0.01</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>0.6</td>
<td>0.09</td>
<td>0.01</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>0.8</td>
<td>0.12</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>KNN</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResNet152</td>
<td>0.2</td>
<td>0.03</td>
<td>0.01</td>
</tr>
<tr>
<td>ImageNet</td>
<td>0.4</td>
<td>0.07</td>
<td>0.01</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>0.6</td>
<td>0.10</td>
<td>0.01</td>
</tr>
<tr>
<td>MNLI</td>
<td>0.8</td>
<td>0.13</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>RoNet18</strong></td>
<td>1.0</td>
<td>0.15</td>
<td>0.01</td>
</tr>
<tr>
<td>ImageNet</td>
<td>0.2</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>0.4</td>
<td>0.07</td>
<td>0.01</td>
</tr>
<tr>
<td>MNLI</td>
<td>0.6</td>
<td>0.10</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>RoBERTs</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MNLJ</td>
<td>1.0</td>
<td>0.07</td>
<td>0.01</td>
</tr>
<tr>
<td>CNN</td>
<td>0.2</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>ResNet28</td>
<td>0.4</td>
<td>0.08</td>
<td>0.01</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>0.6</td>
<td>0.11</td>
<td>0.01</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>0.8</td>
<td>0.14</td>
<td>0.01</td>
</tr>
</tbody>
</table>

21
Table 2: Recalibration Results for Imbalanced Classification. For each dataset and atypicality quantile, the best results are marked in bold. We provide the standard errors next to the means over 10 random seeds.

<table>
<thead>
<tr>
<th>Atypicality</th>
<th>Uncalibrated ECE</th>
<th>Uncalibrated Accuracy</th>
<th>Source</th>
<th>ECE</th>
<th>Accuracy</th>
<th>ECE</th>
<th>Accuracy</th>
<th>ECE</th>
<th>Accuracy</th>
<th>ECE</th>
<th>Accuracy</th>
<th>ECE</th>
<th>Accuracy</th>
<th>ECE</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>0.30 ± 0.02</td>
<td>0.70 ± 0.03</td>
<td>0.42 ± 0.01</td>
<td>0.68 ± 0.01</td>
<td>0.43 ± 0.00</td>
<td>0.41 ± 0.00</td>
<td>0.52 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>0.19 ± 0.01</td>
<td>0.81 ± 0.02</td>
<td>0.34 ± 0.01</td>
<td>0.73 ± 0.01</td>
<td>0.41 ± 0.00</td>
<td>0.40 ± 0.00</td>
<td>0.51 ± 0.00</td>
<td>0.55 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.70</td>
<td>0.17 ± 0.01</td>
<td>0.83 ± 0.02</td>
<td>0.32 ± 0.01</td>
<td>0.75 ± 0.01</td>
<td>0.40 ± 0.00</td>
<td>0.40 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.55 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.80</td>
<td>0.16 ± 0.01</td>
<td>0.84 ± 0.02</td>
<td>0.31 ± 0.01</td>
<td>0.77 ± 0.01</td>
<td>0.39 ± 0.00</td>
<td>0.40 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.55 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.90</td>
<td>0.16 ± 0.01</td>
<td>0.84 ± 0.02</td>
<td>0.31 ± 0.01</td>
<td>0.77 ± 0.01</td>
<td>0.39 ± 0.00</td>
<td>0.40 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.55 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>0.16 ± 0.01</td>
<td>0.84 ± 0.02</td>
<td>0.31 ± 0.01</td>
<td>0.77 ± 0.01</td>
<td>0.39 ± 0.00</td>
<td>0.40 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.55 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td>0.42 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Recalibration Results for LLM Classification. For each dataset and atypicality quantile, the best results are marked in bold. We provide the standard errors next to the means over 10 random seeds.

<table>
<thead>
<tr>
<th>Atypicality</th>
<th>ECE</th>
<th>Uncalibrated RMSE</th>
<th>Uncalibrated Accuracy</th>
<th>ECE</th>
<th>Content-Free RMSE</th>
<th>ECE</th>
<th>Accuracy</th>
<th>ECE</th>
<th>Accuracy</th>
<th>ECE</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>0.30 ± 0.02</td>
<td>0.70 ± 0.03</td>
<td>0.40 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.60 ± 0.00</td>
<td>0.60 ± 0.00</td>
<td>0.60 ± 0.00</td>
<td>0.60 ± 0.00</td>
<td>0.60 ± 0.00</td>
</tr>
<tr>
<td>0.60</td>
<td>0.20 ± 0.01</td>
<td>0.80 ± 0.02</td>
<td>0.30 ± 0.00</td>
<td>0.60 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
</tr>
<tr>
<td>0.70</td>
<td>0.19 ± 0.01</td>
<td>0.81 ± 0.02</td>
<td>0.29 ± 0.00</td>
<td>0.59 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
</tr>
<tr>
<td>0.80</td>
<td>0.18 ± 0.01</td>
<td>0.82 ± 0.02</td>
<td>0.28 ± 0.00</td>
<td>0.58 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
</tr>
<tr>
<td>0.90</td>
<td>0.17 ± 0.01</td>
<td>0.83 ± 0.02</td>
<td>0.27 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.17 ± 0.01</td>
<td>0.83 ± 0.02</td>
<td>0.27 ± 0.00</td>
<td>0.57 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
<td>0.50 ± 0.00</td>
</tr>
</tbody>
</table>
D Recalibration

Through all our recalibration results, we first split the test set into two equally sized calibration and test splits. Then, we fit the recalibration method using the calibration split and compute the performance on the test split. We run all our experiments with 10 different random seeds.

D.1 Temperature Scaling

To perform temperature scaling [GPSW17], we use the calibration set to fit the temperature parameter. To perform the optimization, we use the LBFGS [LN89] algorithm from PyTorch with strong Wolfe line search, following [GPSW17]. Namely, we optimize the parameter \( \tau \)

\[
\hat{P}_\text{TS}(X) = \text{Softmax}(f(X)/\tau)
\]

(9)

and then use it during inference to rescale the logits produced by \( f \). We use 0.1 learning rate and 3000 maximum iterations across all experiments and initialize the temperature value as 1, although find that TS is pretty robust to the choice of hyperparameters.

D.2 Atypicality-Aware Recalibration

Here we describe the implementation details For Atypicality-Aware Recalibration (AAR). We formulate AAR with:

\[
\log \hat{P}_\text{AAR}(Y|X) \propto \phi(a(X)) \log \hat{P}(Y|X) + S_Y,
\]

(10)

In total, this gives us \(|\mathcal{Y}| + 3\) parameters. Using exactly the same setting as TS, we use LBFGS with a strong Wolfe search to optimize the three parameters, with the same splits as temperature scaling. We normalize the atypicality values (subtract the mean and divide by the standard deviation of the calibration set) for numerical stability. We use the same hyperparameters as TS (with 0.1 learning rate and 3000 maximum iterations) without any modification across all experiments, initialize \( c_0, c_1, c_2 \) as 0 and \( S_Y \) parameters as 1. We run the recalibration procedure on a CPU with precomputed logits.

D.2.1 Adaptive Temperature Scaling

For AdaTS [JPL+23] we use the implementation provided with the paper 7. We identically use the hyperparameters and the architecture provided in the paper and their repository. They use an encoder and decoder architecture with \([1024, 512, 512]\) hidden units each, and a temperature predictor network with \([128, 128]\) hidden units. They use an Adam Optimizer with a learning rate of \(5e^{-4}\) with \(128\) batch size.

D.3 Conformal Prediction

We follow the presentation in [AB21, ABMJ20]. Let \( \pi(X) \) be the permutation of \( \mathcal{Y} = \{1, \ldots, C\} \) that sorts \( \hat{P}(Y = c|X) \), i.e. the predicted probabilities for each class \( c \). We define a score function

\[
s(x, y) = \sum_{j=1}^{c} \hat{P}(Y = j|X), \text{ where } y = \pi_c.
\]

(11)

This means greedily including classes until the set contains the true label, and using the cumulative sum of the probabilities as the score function. We compute all of the scores for the calibration set, \( S_{\text{calib}} = \{s(x_1, y_1), \ldots, s(x_N, y_N)\} \), we the \( \frac{(N+1)(1-\alpha)}{N} \)-th quantile of the scores, \( \hat{q} \). Then, the prediction set is defined as

\[
C(x) = \{y : s(x, y) \leq \hat{q}\}
\]

(12)

We can further add randomization to the procedure where we have the prediction set function to be \( C(x, u) : \mathcal{X} \times [0,1] \) for randomization purposes to satisfy exact coverage. We refer to [VGS05, ABMJ20, AB21] for a more thorough presentation.

---

7https://github.com/thwjoy/adats
RAPS is a variant of APS that regularizes the set sizes. They modify the scoring function to add a regularization term. This is controlled by the test size offset $k_{\text{reg}}$ that controls the value beyond which the regularization is applied, and the $\lambda_{\text{reg}}$ gives the strength of the regularization. To fit the $k_{\text{reg}}, \lambda_{\text{reg}}$ parameters in RAPS, we follow the procedure in [ABMJ20] to fit both parameters. Namely, we fit $k_{\text{reg}}$ by Algorithm 4 in their paper that leverages the set sizes in the calibration set, and we fit $\lambda_{\text{reg}}$ by the largest regularization parameter that achieves the smallest set sizes, searched over a grid of \{0.001, 0.01, 0.1, 0.2, 0.5\} following their presentation.

D.4 Atypicality-Aware Conformal Prediction

We have a simple discrete grouping scheme to make conformal prediction atypicality aware. Namely, we group points using their atypicality and confidence percentiles and fit individual thresholds. Concretely, we construct a dataset of $D_{AA} = \{(c_i, c_{i+1}], (a_j, a_{j+1}], \hat{q}_{i,j}\}_{i,j\in[N]}$ using the calibration set where $(c_i, c_{i+1}]$ denotes the confidence range for quantile $i$, $(a_j, a_{j+1}]$ denotes the atypicality range for quantile $j$ and $\hat{q}_{i,j}$ denotes the threshold fitted to the group specified by these intervals. We let $N = 6$ as the number of groups, and in total, we end up with 36 thresholds. At test time, we check the quantile of the confidence and atypicality of a point and use the corresponding temperature. For AA-RAPS, we use the same $k_{\text{reg}}$ and $\lambda_{\text{reg}}$ values as was found with the RAPS procedure. For practical purposes, we do not allow zero sets (prediction sets at least include the top prediction).

We would like to make the remark that sometimes the marginal coverage can exceed the desired value (e.g. Figure 9). This is often because the underlying model is already very confident for a majority of data points (e.g. More than half of the data points have 92% confidence). The gains we provide are often for points with lower confidence regions, as the coverage is not satisfied in those regions.
Figure 8: **Atypicality-Aware Conformal Prediction for ResNet18 and ImageNet.** Target coverage rate is 95%.

Figure 9: **Atypicality-Aware Conformal Prediction for ResNet152 and ImageNet.** Target coverage rate is 95%.

Figure 10: **Fitted Temperature vs Atypicality.** We observe a monotonically increasing relationship between the atypicality of a group and the temperature parameter fitted to that group with TS.

### E Tables for Results

Here, we present the table version of the results in Figure 3. Tables 1,2 contain the ECE analysis.
F Fitzpatrick17k and Skin Lesion Classification

We use the training script from [GHS+21] to finetune models on the Fitzpatrick17k dataset. We train the models for 50 epochs, fixing the backbone and training only the probe on top of the penultimate layer. The probe consists of 2 layers, one layer of 256 units followed by ReLU and Dropout with probability 0.4, followed by the classifier layer with an output dimensionality of 9. We use an Adam optimizer with a 0.0001 learning rate.

The entire dataset consists of 16,577 images, where the potential labels are: 10,886 inflammatory, 1,352 malignant epidermal, 1,194 genodermatoses, 1,067 benign dermal, 931 benign epidermal, 573 malignant melanoma, 236 benign melanocyte, 182 malignant cutaneous lymphoma, and 156 malignant dermal. We split the dataset into 3 sets (Training (0.5), Validation (0.25), and Test (0.25)). We use the validation set as the calibration set and perform the experiments with 10 random splits.
G Proofs

G.1 Detailed derivation of the claim on Page 5

When \( \mathbb{P}_1(X) \leq 1/2 \), the signed calibration error at level \( u \in (1/2, 1) \) becomes \( u - \mathbb{P}(Y = -1 \mid \mathbb{P}_1(X) = u) = u - \mathbb{P}(Y = -1 \mid \mathbb{P}_1(-X) = u) = u - \mathbb{P}(Y = 1 \mid \mathbb{P}_1(X) = u) \).

The last inequality is due to symmetry. More specifically, we claim \( (X, Y) \overset{d}{=} (-X, -Y) \), where the notation \( \overset{d}{=} \) denotes equal in distribution. In fact, as \( X = -X \), it suffices to show that for any \( y \in \{-1, 1\} \), and \( x \in \mathbb{R}^d \), we have

\[
\mathbb{P}(Y = y \mid X = x) = \mathbb{P}(-Y = y \mid -X = x).
\]

When \( y = -1 \), the right hand side

\[
\mathbb{P}(-Y = -1 \mid -X = x) = \mathbb{P}(Y = 1 \mid X = -x) = \sigma(\langle \beta^*, -x \rangle)
\]

\[
= 1 - \sigma(\langle \beta^*, x \rangle) = 1 - \mathbb{P}(Y = 1 \mid X = x) = \mathbb{P}(Y = -1 \mid X = x).
\]

Similarly, when \( y = 1 \),

\[
\mathbb{P}(-Y = 1 \mid -X = x) = \mathbb{P}(Y = -1 \mid X = -x) = 1 - \sigma(\langle \beta^*, -x \rangle)
\]

\[
= \sigma(\langle \beta^*, x \rangle) = \mathbb{P}(Y = 1 \mid X = x).
\]

We complete the proof.

G.2 Proof of Theorem 3.1

**Theorem G.1 (Restatement of Theorem 3.1).** Consider the data generative model with the algorithm described in Section 3.2. For any \( K > 1 \), suppose we consider the quantiles of \( a(X) \), \( a_1, a_2, \ldots, a_K, a_{K+1} \) such that \( \mathbb{P}(a(X) \in (a_k, a_{k+1}]) = 1/K \) for \( k \in [K] \). In addition, we assume \( \|\beta^*\| \leq c_0 \), and \( d/n = \kappa \) for some sufficiently small \( c_0, \kappa > 0 \). Then for sufficiently large \( n \), we have

\[
\mathbb{E}_u [u - \mathbb{P}(Y = 1 \mid \hat{P}_1(X) = u) \mid a(X) \in [a_{k-1}, a_k]] >
\]

\[
\mathbb{E}_u [u - \mathbb{P}(Y = 1 \mid \hat{P}_1(X) = u) \mid a(X) \in (a_k, a_{k+1})],
\]

for \( k = 2, \ldots, K \).

**Proof.** Following [BMWX21a], we have

\[
u - \mathbb{P}(Y = 1 \mid \hat{P}_1(X) = u) = u - \mathbb{E}_Z \sigma(\|\beta^*\| \cos \hat{\theta} \cdot \sigma^{-1}(u)) + \sin \hat{\theta} \cdot \|\beta^*\| Z,
\]

where \( \cos \hat{\theta} = \frac{\hat{\beta}^T \beta^*}{\|\beta^*\|} \) and \( Z \sim N(0, 1) \).

According to the results in Section 2.2 of [SC19], we have \( \|\hat{\beta}\| \to R^* = R^*(\kappa, \beta^*) \) and \( \cos \hat{\theta} \to c^* = c^*(\kappa, \beta^*) \), for two quantities \( R^* \) and \( c^* \) that depend on \( \kappa \) and \( \beta^* \). We then have

\[
u - \mathbb{P}(Y = 1 \mid \hat{P}_1(X) = u) \to u - \mathbb{E}_Z \sigma(\|\beta^*\| \frac{c^*}{R^*} \sigma^{-1}(u)) + \sqrt{1 - c^*^2} \cdot \|\beta^*\| Z.
\]

Using the proof of Theorem 3 in [BMWX21a], we have that

\[
u - \mathbb{P}(Y = 1 \mid \hat{P}_1(X) = u) = C_{\kappa}(u) \cdot \kappa + o(\kappa),
\]

where

\[
C_{\kappa}(u) = c_1 \sigma'(\sigma^{-1}(u)) \cdot \sigma^{-1}(u) - c_2 \sigma''(\sigma^{-1}(u)),
\]

for two positive constants \( c_1, c_2 \).

As a result, we have

\[
u - \mathbb{P}(Y = 1 \mid \hat{P}_1(X) = u) \geq 0 \tag{13}
\]

In addition, since when \( z \in [-1, 1] \), \( z \cdot \sigma'(z) \) and \( -\sigma''(z) \) are both increasing, we then have \( C_{\kappa}(u) \) increasing for \( \hat{\beta}^T x = \sigma^{-1}(u) \in (-1, 1) \).
We now use the approximation of the chi-square quantile: when
\[ p = a \cdot \frac{\| \beta \|}{\| S \|} \]
where \( S \) is the first coordinate of \( S \).

Consequently, if we further condition on the event where \( \beta^T x > 0 \) (as we assume \( u > 0 \) throughout Section 3.2), we have
\[
\hat{\beta}^T x = a \cdot \frac{\| \hat{\beta} \|}{\| S \|} \cdot S_1 > \frac{d}{2} \cdot \frac{Z_1}{\sqrt{Z_1^2 + Q}} \to a \cdot R^* \cdot \frac{Z_1}{\sqrt{Z_1^2 + Q}}.
\]

Due to the monotonicity of \( C_\alpha(u) \) on \( u \), we have that for any \( a_1 > a_2 \),
\[
C_\alpha(u) \mid \| x \| = a_1 > C_\alpha(u) \mid \| x \| = a_2,
\]
where the notation \( > \) denotes stochastic dominance.

Consequently, we have
\[
E_u[u - P(\hat{Y}_1) \mid \hat{P}_1(X) = u) \mid a(X) \in [a_{k-1}, a_k)] < E_u[u - P(\hat{Y}_1) \mid \hat{P}_1(X) = u) \mid a(X) \in (a_k, a_{k+1})],
\]
for \( k = 2, \ldots, K - 1 \).

Proving the result for \( k = K \)

To complete the proof, it suffices to show that the inequality is also true for \( \chi^2 \) quantile:
\[
E_u[u - P(\hat{Y}_1) \mid \hat{P}_1(X) = u) \mid a(X) \in [a_{k-1}, a_K)] < E_u[u - P(\hat{Y}_1) \mid \hat{P}_1(X) = u) \mid a(X) \in (a_k, a_{k+1})],
\]
which is equivalent to
\[
E_u[(u - P(\hat{Y}_1) = 1) \mid \hat{P}_1(X) = u) \cdot 1\{a(X) \in [a_{k-1}, a_K]\}] < E_u[(u - P(\hat{Y}_1) = 1) \mid \hat{P}_1(X) = u) \cdot 1\{a(X) \in (a_k, a_{k+1})\}].
\]

In the above inequality, the right hand side can be decomposed into
\[
E_u[(u - P(\hat{Y}_1) = 1) \mid \hat{P}_1(X) = u) \cdot 1\{a(X) \in (a_k, a_{k+1})\}]
= E_u[(u - P(\hat{Y}_1) = 1) \mid \hat{P}_1(X) = u) \cdot 1\{a(X) \in [a_k, 2p]\}] + E_u[(u - P(\hat{Y}_1) = 1) \mid \hat{P}_1(X) = u) \cdot 1\{a(X) \in [2p, a_{K+1})\}].
\]

Denote the \( \alpha \) quantile of \( \chi^2_p \) by \( \chi^2_{\alpha,p} \). We then have \( a_k = \chi^2_{\alpha, p} \cdot \frac{1}{n^{1/p}} \). We further decompose the equation into
\[
E_u[(u - P(\hat{Y}_1) = 1) \mid \hat{P}_1(X) = u) \cdot 1\{a(X) \in [a_k, 2p]\}] = E_u[(u - P(\hat{Y}_1) = 1) \mid \hat{P}_1(X) = u) \cdot 1\{a(X) \in [a_k, \chi^2_{\alpha, p} \cdot \frac{1}{n^{1/p}}]\}]
+ E_u[(u - P(\hat{Y}_1) = 1) \mid \hat{P}_1(X) = u) \cdot 1\{a(X) \in [\chi^2_{\alpha, p} \cdot \frac{1}{n^{1/p}}, 2p]\}].
\]

In the following, we proceed to prove
\[
E_u[(u - P(\hat{Y}_1) = 1) \mid \hat{P}_1(X) = u) \cdot 1\{a(X) \in [\chi^2_{\alpha, p} \cdot \frac{1}{n^{1/p}}, 2p]\}] > E_u[(u - P(\hat{Y}_1) = 1) \mid \hat{P}_1(X) = u) \cdot 1\{a(X) \in [a_{k-1}, a_K]\}].
\]

(14)

We now use the approximation of the chi-square quantile: when \( p \to \infty \), we have
\[
a_k = \frac{1}{2} (z_{k, \frac{1}{n^{1/p}}} + \sqrt{2p})^2 + o(1), \quad \text{and} \quad \chi^2_{\alpha, p} = \frac{1}{2} (z_{\alpha, \frac{1}{n^{1/p}}} + \sqrt{2p})^2 + o(1),
\]

28
where \( z_\alpha \) denotes the \( \alpha \)-quantile of a standard normal random variable.

Then
\[
\chi^2_{K+1,p} - a_K = \frac{1}{2} \left( (z_{K+1} - z_K)(z_{K+1} + z_K) + 2\sqrt{2p} \right).
\]

Using the fact that \( z_{1-K^{-\delta}} = \sqrt{2\log K} + o(1) \) for \( K \to \infty \), then we have
\[
z_{K+1} - z_K = -\frac{\log(1-\delta)}{\sqrt{2\log K}} + o(1).
\]

In addition, for any \( a \in [\chi^2_{K+1,p}, 2p] \) and \( a' \in [a_{K-1}, a_K] \), we have
\[
\mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) | a(X) = a] - \mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) | a(X) = a'] \geq C(z_{K+1} - z_K),
\]
for some universal constant \( C \).

Therefore
\[
\mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) | a(X) \in [\chi^2_{K+1,p}, 2p]] - \mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) | a(X) \in [a_{K-1}, a_K]] \geq C(z_{K+1} - z_K).
\]

Then
\[
\mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) \cdot 1\{a(X) \in [\chi^2_{K+1,p}, 2p]\}] = \mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) | a(X) \in [\chi^2_{K+1,p}, 2p] \cdot P(a(X) \in [\chi^2_{K+1,p}, 2p])
\geq \left( \mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) | a(X) \in [a_{K-1}, a_K]] + C(z_{K+1} - z_K) \right) \cdot \left( \frac{1}{K} - \frac{\delta}{K+1} + o(\frac{\delta}{K+1}) \right)
= \mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) \cdot 1\{a(X) \in [a_{K-1}, a_K]\}
+ C(z_{K+1} - z_K) - (1 + o(1)) \cdot \frac{\delta}{K+1} \cdot \mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) | a(X) \in [a_{K-1}, a_K]].
\]

The last equality uses the fact that \( P(a(X) \in [a_{K-1}, a_K]) = 1/K \), and therefore
\[
\mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) | a(X) \in [a_{K-1}, a_K]] = \frac{1}{K} = \mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) \cdot 1\{a(X) \in [a_{K-1}, a_K]\}]
\]

Then use the fact that \( |\mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) | a(X) \in [a_{K-1}, a_K]] = O(1) \) and we choose \( \delta = o(1/\log K) \) so
\[
\frac{\delta}{K} = o\left(\frac{\log(1-\delta)}{\sqrt{\log K}}\right).
\]

Consequently,
\[
C(z_{K+1} - z_K) - (1 + o(1)) \cdot \frac{\delta}{K+1} \cdot \mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) | a(X) \in [a_{K-1}, a_K]] > 0,
\]
which implies
\[
\mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) \cdot 1\{a(X) \in [\chi^2_{K+1,p}, 2p]\}] \geq \mathbb{E}_u[(u - P(Y = 1 | \hat{P}_1(X) = u)) \cdot 1\{a(X) \in [a_{K-1}, a_K]\}].
\]

Combining with equation 13, we prove equation 14 and complete the proof. \( \square \)
G.3 Theoretical Justification of the calibration improvement using the atypicality score

In this section, we provide the theoretical justification to understand why incorporating the atypicality score will improve calibration. In particular, we consider the binary classification problem with prediction \( f : X \in [0, 1] \) indicating the predicted probability of \( Y = 1 \) given \( X = x \).

For a predictor \( f \), let us denote its conditional calibration error at an atypicality level \( \gamma \) by \( CE(\gamma) = E[(f(X) - Y)^2 \mid a(X) = \gamma] \).

**Theorem G.2.** Consider the same setting as Theorem 3.1. Suppose the temperature function \( \hat{\tau}(a(X)) = \arg \min \tau \ E[l(Y, \text{Softmax}(f(X)/\tau(a(X))))] \) with \( l \) being the cross entropy loss, and let \( \hat{\tau}_AAR(X) = \text{Softmax}(f(X)/\hat{\tau}(a(X))) \). Then

\[
CE(\hat{\tau}_{AAR}) \leq \min \{ CE(\hat{\tau}_{TS}), CE(\gamma) \}. \tag{15}
\]

Proof: For a prediction function \( f \), we first define the conditional mean squared error of \( f \) at an atypicality level \( \gamma \) by \( MSE(\gamma) = E[(f(X) - Y)^2 \mid a(X) = \gamma] \), then we have

\[
MSE(\gamma) - CE(\gamma) = E[(f(X) - Y)^2 \mid a(X) = \gamma] - E[(f(X) - E[Y \mid f(X), a(X) = \gamma])^2 \mid a(X) = \gamma]
\]

\[
= E[(E[Y \mid f(X), a(X) = \gamma] - Y) \cdot (f(X) - E[Y \mid f(X), a(X) = \gamma] - Y) \mid a(X) = \gamma]
\]

\[+ 2E[(E[Y \mid f(X), a(X) = \gamma] - Y) \cdot (f(X) - E[Y \mid f(X), a(X) = \gamma]) \mid a(X) = \gamma]
\]

Since

\[
E[Y \mid f(X), a(X) = \gamma] - Y \cdot (f(X) - E[Y \mid f(X), a(X) = \gamma]) \mid a(X) = \gamma = 0,
\]

and therefore

\[
MSE(\gamma) - CE(\gamma) = E[(E[Y \mid f(X), a(X) = \gamma] - Y)^2 \mid a(X) = \gamma]
\]

Now that \( \hat{\tau}_{AAR}(f(x), a(x)) \) is monotonic on the \( \hat{P}(x) \), we have

\[
E[Y \mid f(x), a(X) = \gamma] = E[Y \mid \hat{\tau}_{AAR}(f(x), a(x)), a(X) = \gamma],
\]

implying

\[
MSE(\gamma) - CE(\gamma) = MSE(\gamma) - CE(\gamma) \tag{16}
\]

Similarly, we have

\[
MSE(\gamma) - CE(\gamma) = MSE(\gamma) - CE(\gamma) \tag{17}
\]

In the following, we will show that

\[
MSE(\gamma) - CE(\gamma) < \min \{ MSE(\gamma), MSE(\gamma) \}. \tag{18}
\]

First, as we consider the binary classification setting, with \( l \) being the cross-entropy loss, we have

\[
l(Y, \text{Softmax}(f(X)/\tau(a(X)))) = Y \log(\sigma(f(X)/\tau(a(X)))) + (1 - Y) \log(1 - \sigma(f(X)/\tau(a(X))),
\]

where \( \sigma(x) = 1/(1 + e^x) \).

Then, by the definition of \( \hat{\tau}(a(X)) \), we have that

\[
\hat{\tau}(a(X)) = \arg \min \tau E[Y \log(\sigma(f(X)/\tau(a(X)))) + (1 - Y) \log(1 - \sigma(f(X)/\tau(a(X))))]
\]

\[= \arg \min \tau E[\E[Y \log(\sigma(f(X)/\tau(a(X)))) + (1 - Y) \log(1 - \sigma(f(X)/\tau(a(X)))) \mid a(X)]].
\]
Taking the derivative on the last line and setting it to zero, we have
\[
E\left[ \frac{Y}{\sigma(f(X)/\hat{\tau}(a(X)))} - \frac{1-Y}{1-\sigma(f(X)/\hat{\tau}(a(X)))} \mid a(X) \right] = 0,
\]
implying
\[
E[\sigma(f(X)/\hat{\tau}(a(X))) \mid a(X)] = E[Y \mid a(X)].
\]
This makes the derivative of \(E[(Y - \sigma(f(X)/\tau(a(X))))^2 \mid a(X)]\) zero and therefore \(\hat{\tau}(a(X))\) is also a minimizer of
\[
E[(Y - \sigma(f(X)/\tau(a(X))))^2 \mid a(X)] = \arg\min_{\tau} E[(Y - \sigma(f(X)/\tau(a(X))))^2]
\]
Letting \(g(\gamma) = \arg\min_c E[(Y - \sigma(f(X)/c))^2 \mid a(X) = \gamma]\), we have that
\[
g(a(X)) = \arg\min_{\tau} E[(Y - \sigma(f(X)/\tau(a(X))))^2 \mid a(X)] = \hat{\tau}(a(X)).
\]
and therefore
\[
g(a(X)) = \arg\min_{\tau} E[(Y - \sigma(f(X)/\tau(a(X))))^2 \mid a(X)] = \hat{\tau}(a(X)).
\]
As a result,
\[
\text{MSE}_x(\hat{P}_{\text{AAR}}) = E[(\hat{P}_{\text{AAR}}(X) - Y)^2 \mid a(X) = \gamma] = E[(\hat{P}_{\text{AAR}}(X) - Y)^2 \mid a(X) = \gamma] = E[(\sigma(f(X)/\hat{\tau}(a(X)) - Y)^2 \mid a(X) = \gamma] = E[(\text{Softmax}(\hat{P}(X)/g(a(X)) - Y)^2 \mid a(X) = \gamma] = \arg\min_{c} E[(\sigma(f(X)/c - Y)^2 \mid a(X) = \gamma] \leq E[(\sigma(f(X) - Y)^2 \mid a(X) = \gamma] = \text{MSE}_x(\hat{P}).
\]
Similarly, we have \(\text{MSE}_x(\hat{P}_{\text{AAR}}) \leq \text{MSE}_x(\hat{P}_{\text{TS}})\), and therefore equation 18 holds.
Combining with equation 16 and equation 17, we have
\[
CE_{\gamma}(\hat{P}_{\text{AAR}}) \leq \min\{CE_{\gamma}(\hat{P}_{\text{TS}}), CE_{\gamma}(\hat{P})\}.
\]
<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>Input Atypicality Group</th>
<th>Group Coverage</th>
<th>APS Softmax</th>
<th>Coverage</th>
<th>AA-APS Softmax</th>
<th>Coverage</th>
<th>RAPS Softmax</th>
<th>Coverage</th>
<th>AA-RAPS Softmax</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet50</td>
<td>ImageNet 1</td>
<td>1</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td>ResNet50</td>
<td>ImageNet 1</td>
<td>1</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>0.96e+00</td>
<td>0.570</td>
<td>0.96e+00</td>
<td>0.183</td>
<td>0.95e+00</td>
<td>0.217</td>
<td>0.94e+00</td>
<td>0.204</td>
<td>0.94e+00</td>
</tr>
</tbody>
</table>

Table 4: Conformal Calibration with Atypicality-Awareness.
H Limitations

H.1 Quantifying Atypicality

Since we do not have access to the true distribution of $P(X)$, we estimate it through the model, e.g. using the embeddings. This means we are capturing the atypicality not solely with respect to the training distribution but also the model. It is possible that a model that does not fit the data well and produces low-quality atypicality estimates. We would like to stress that our goal here is to show that even simple estimators can demonstrate significant benefits. In general, we observe that our findings hold for large datasets and widely used models, and atypicality gives a semantically meaningful way to group data points qualitatively. Our findings suggest that we can unify the understanding and improve uncertainty quantification and recalibration methods with atypicality, however, practitioners should be careful about incorporating atypicality, as poor atypicality estimates can lead to worse performance.

H.2 Subgroup Fairness Experiments

While the literature on algorithms to satisfy group fairness is rich [KGZ19, HJKRR18], here we wanted to give a case study with skin-lesion classification. Our goal was to provide further evidence that atypicality awareness could improve fairness algorithms. In this domain, there is more verification to do to better characterize how and when atypicality helps, thus, to better understand which subgroups could benefit more from atypicality-aware algorithms and whether these findings apply generally in the literature.

H.3 Theoretical Analysis

Following the earlier work [BMWX21a, SC19], we analyzed the calibration behavior of well-specified logistic regression. However, our empirical findings suggest that the phenomena are much more broadly applicable. We suggest that future work can analyze the behavior in more general settings to better understand the dynamics.
Figure 14: **Distribution of Input Atypicality.** Here, we give the distribution of Atypicality for ResNet18 on ImageNet, using GMM and KNN methods.

Figure 15: **Label Atypicality and Confidence.** Here, the x-axis reflects the label atypicality quantile, and the y-axis indicates confidence. The coloring for the figure on the left indicates the accuracy within a bin, and the figure on the right has the difference between confidence and accuracy within a bin. Similar to Figure 6, we observe that atypical examples have lower accuracy, and predictions are more overconfident.

Figure 16: **Input Atypicality and Confidence for ResNet152.** We provide the input atypicality for ResNet152, in the same structure as Figure 15.