# BEYOND HARD DECISIONS: ACCOUNTING FOR UNCERTAINTY IN DEEP MIR MODELS

**Qingyang Xi**
New York University
tom.xi@nyu.edu

**Brian McFee**
New York University
brian.mcfee@nyu.edu

## ABSTRACT

Modern deep learning models provide increasingly more accurate predictions for common MIR tasks, however, the models' confidence scores associated with each prediction are often left unchecked. This potential mismatch between prediction confidence and empirical accuracy makes it difficult to account for uncertainties in these models' predictions. Controlling uncertainty is crucial if MIR models' prediction confidences are to be interpreted as probabilities, and doing so can help a model produce more meaningful predictions when faced with ambiguity. To properly account for model uncertainties, prediction confidence scores should be calibrated to better reflect the true chance of it being correct. We propose a simple and efficient post-hoc probability calibration process using Temperature Scaling. We demonstrate the effect of this calibration process on the Rock Corpus for key and chord estimation.

## 1. INTRODUCTION AND DEFINITIONS

Modern deep neural-network models have been able to achieve impressive accuracy on many music information retrieval (MIR) tasks [1–3]. However, typical models don't properly account for uncertainties in their predictions, and result in a mismatch between the model's reported confidence and its empirical accuracy [4]. Calibrated prediction confidence is necessary for incorporating model outputs into probabilistic models.

Deep MIR models usually produce a confidence score $\hat{p}$ associated with a prediction $\hat{y}$ by applying a softmax function to the network output layer $\mathbf{z} \in \mathbb{R}^K$, where

$$\hat{\mathbf{p}} = \sigma_{\mathrm{SM}}(\mathbf{z}), \quad \hat{y} = \arg\max_k \hat{\mathbf{p}}^{(k)}, \quad \hat{p} = \max_k \hat{\mathbf{p}}^{(k)}$$

and the softmax function $\sigma_{\mathrm{SM}}$ is defined as

$$\sigma_{\mathrm{SM}}(\mathbf{z}) = \frac{\exp(\mathbf{z})}{\sum_{k=1}^{K} \exp(\mathbf{z}^{(k)})}$$

Consider a K-class classifier $h(\mathbf{x}) = (\hat{y}, \hat{p})$ with input feature $\mathbf{x}$, class prediction $\hat{y} \in \{1, \ldots, K\}$, and prediction confidence $\hat{p} \in [0, 1]$. To have calibrated confidence scores means that $\hat{p}$ should represent how often the classifier produces a correct prediction $\hat{y} = y$ for inputs $\mathbf{x}$ from $(\mathbf{x}, y) \sim \mathcal{D}$. Formally, perfect calibration is defined as

$$\mathbb{P}(\hat{y} = y | X = \mathbf{x}) = \hat{p}(\mathbf{x}) \tag{1}$$

which equates the accuracy of the model with the model confidence score with regard to the same input. We estimate the left hand side of Eqn (1) using the average model accuracy over members of a dataset $(\mathbf{x}_i, y_i)_{i=1}^N \sim \mathcal{D}$.

While perfect calibration is often impossible to achieve, we can measure how reliable a model is by estimating the gap between the left hand side and right hand side of Eqn (1). We can do that by looking at the confidence histogram (figure 1, right). If we group inputs $\mathbf{x}_i$ by which histogram bin $\hat{p}(\mathbf{x}_i)$ falls into, we can calculate both the empirical accuracy and average confidence of that group, which estimates the left and right-hand-side of equation (1) respectively.

A reliability diagram (figure 1, left) plots the empirical accuracy as a function of confidence, and shows how well calibrated a model is [5, 6]. If a model is perfectly calibrated, then the reliability diagram should resemble identity function; any mis-calibration would result in a gap between empirical accuracy and confidence.

Another more compact metric of calibration is the Expected Calibration Error (ECE), which is a scalar summary of the reliability diagram [7]. It measures the expected absolute difference between prediction confidence and accuracy and is defined as the average of the accuracy confidence gap (dashed boxes in figure 1), weighted by the number of points landing in the corresponding bin (Figure 1 right).

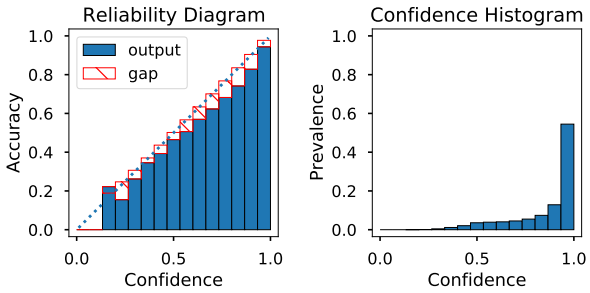


**Figure 1**. Reliability Diagram (left) and confidence histogram (right) of the CREMA root predictor on its test set.

## 2. CALIBRATION TECHNIQUE

The goal of probability calibration is to produce a calibrated confidence $\hat{q}$, based on observations. Many techniques for calibrating prediction confidence exist for multi-class models [7–9]. We choose temperature scaling due to its effectiveness on a wide range of tasks and simplicity in implementation [4]. Temperature calibration uses a single parameter $\beta > 0$ to scale the logit vector $\mathbf{z}$, and produces the calibrated confidence:

$$\hat{q} = \max_k \hat{\mathbf{q}}^{(k)} = \max_k \sigma_{\mathrm{SM}} \left(\beta \cdot \mathbf{z}\right)^{(k)} \tag{2}$$

Since $\beta$ is always positive, temperature scaling does not change $\arg\max_k \sigma_{\mathrm{SM}} \left(\beta \cdot \mathbf{z}\right)^{(k)}$, which is the model's original prediction $\hat{y}$.

While the original formulation of Guo et al. [4] requires direct access to the model logits $\mathbf{z}$, we propose using $\log \hat{\mathbf{p}}$ as a proxy for $\mathbf{z}$ where $\hat{\mathbf{p}} = \sigma_{\mathrm{SM}}(\mathbf{z})$ is the vector of class likelihoods. This allows us to apply the calibration process post-hoc on trained models, where logits are not readily accessible. Letting $\mathbf{z} = \log \hat{\mathbf{p}}$, we observe:

$$\hat{\mathbf{q}} = \sigma_{\mathrm{SM}} \left(\beta \log \hat{\mathbf{p}}\right) = \sigma_{\mathrm{SM}} \left(\log \hat{\mathbf{p}}^{\beta}\right) = \frac{\hat{\mathbf{p}}^{\beta}}{\sum_k \left(\hat{\mathbf{p}}^{\beta}\right)^{(k)}} \tag{3}$$

This provides a straightforward recipe for obtaining calibrated probability for each class after the calibration constant $\beta$ has been determined on a calibration set: simply raise the model softmax output to $\beta$ and normalize.

The choice of $\beta$ is optimized with respect to the cross entropy between the calibrated distribution $\hat{\mathbf{p}}(\mathbf{x})$ and a labeled calibration set, $(\mathbf{x}_i, y_i)_{i=1}^N \sim \mathcal{D}$.

$$\beta^* = \arg\min_{\beta} -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\left(\hat{\mathbf{p}}^{\beta}\right)^{(y_i)}}{\sum_k \left(\hat{\mathbf{p}}^{\beta}\right)^{(k)}} \tag{4}$$

This is a convex optimization problem over a single variable $\beta$, and can be readily solved by most numerical optimization toolkits.

## 3. EXPERIMENTS AND RESULTS

We tested the effect of temperature calibration on two popular deep MIR models: CREMA's Chord root estimator [3] and MADMOM's Key estimator [10]. We used CREMA's test set as the calibration set for the chord root estimator, and both the GiantSteps Key [11] and the Billboard dataset [12] as the calibration set for the key estimator. Entries in GiantSteps Key and Billboard that have modulations or labels outside of MADMOM's key vocabulary are discarded. Both the GiantSteps Key and Billboard set are obtained via MirData [13]. Since MADMOM is designed to produce a single key for each excerpt, songs in the Rock Corpus are broken up into segments of single key before being analyzed by the key model. Segments that are shorter than 5 seconds are discarded.

After calibration, we combine the estimated chord root and key centers to produce an estimate of which scale degree the chord is rooted upon, what we call relative roots. (a chord root of G in the key of D has relative root of IV.)

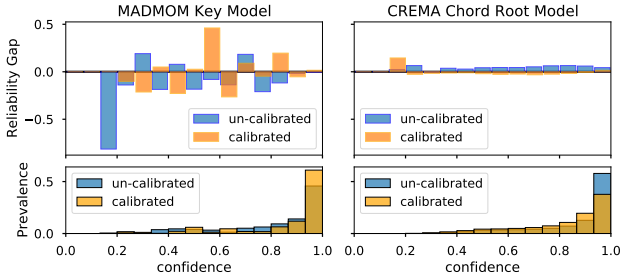


**Figure 2**. Shifts in reliability gaps (top) and confidence histograms (bottom) for applying calibration to the key estimator (left) or the chord estimator (right) individually.

| Key \ Root | H | U | C | A |
|---|---|---|---|---|
| H | 23.66% | 11.44% | 6.18% | 13.65% |
| U | 10.16% | 4.55% | 7.07% | 6.74% |
| C | 13.03% | 2.21% | 3.64% | 5.99% |
| A | 17.00% | 4.78% | 1.59% | 0% |

**Table 1**. Expected Calibration Error (ECE) of the realtive root analysis produced by using one of four outputs (C: calibrated, U: un-calibrated, H: hard decisions, A: annotation) for either the key or the chord root model.

We show the effect of calibrating the constituent estimators on the resulting combined analysis via an ablation study, using either the calibrated output, the un-calibrated output, the hard decision, or the annotation (oracle) for the key and chord root predictors respectively to produce 15 relative root analyses (and 1 annotation) of the Rock Corpus, with their ECEs recorded in Table 1.

By minimizing the objective function Eqn (4) over the respective calibration sets using the bounded Brent's method [14] in the `scipy.optimize` package [15], we found that $\beta^* = 0.79$ for CREMA, and $\beta^* = 1.25$ for MADMOM. Figure 2 shows the individual effect of calibrating either the key or the chord model on the reliability of their respective tasks. Both models show an improvement in ECE: from 4.78% to 1.59% for CREMA (Table 1 last row), and from 6.74% to 5.99% for MADMOM (Table 1 last column).

It is curious that while calibrating the root model improved ECE when paired with deterministic key predictions (row 1 and 4 of Table 1), they didn't help when paired with probabilistic key outputs (row 2 and 3). Nevertheless, the analysis using calibrated outputs outperforms the analysis using hard decisions (diagonal of Table 1).

## 4. CONCLUSION AND FUTURE WORK

By calibrating the output of two deep MIR models, we demonstrate the potential of temperature calibration, which effectively improves the reliability of the analysis in general. Given the simplicity in the calibration technique and its ease of implementation, it can be readily incorporated into an MIR pipeline.

## 5. REFERENCES


[1] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," *Proc. International Society for Music InformationRetrieval Conference, Paris*, 2018.

[2] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 161–165.

[3] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition." in *ISMIR*, 2017, pp. 188–194.

[4] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[5] M. H. DeGroot and S. E. Fienberg, "The comparison and evaluation of forecasters," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 32, no. 1-2, pp. 12–22, 1983.

[6] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 625–632.

[7] M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

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

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

[10] F. Korzeniowski and G. Widmer, "Genre-agnostic key classification with convolutional neural networks," *arXiv preprint arXiv:1808.05340*, 2018.

[11] P. Knees, Á. Faraldo Pérez, H. Boyer, R. Vogl, S. Böck, F. Hörschläger, M. Le Goff *et al.*, "Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR); 2015 Oct 26-30; Málaga, Spain.[Málaga]: International Society for Music Information Retrieval, 2015. p. 364-70.* International Society for Music Information Retrieval (ISMIR), 2015.

[12] J. A. Burgoyne, J. Wild, and I. Fujinaga, "An expert ground truth set for audio chord recognition and music analysis." in *ISMIR*, vol. 11, 2011, pp. 633–638.

[13] R. M. Bittner, M. Fuentes, D. Rubinstein, A. Jansson, K. Choi, and T. Kell, "mirdata: Software for reproducible usage of datasets," in *ISMIR*, 2019.

[14] R. P. Brent, "Chapter 4: An algorithm with guaranteed convergence for finding a zero of a function," in *Algorithms for Minimization without Derivatives*. Prentice-Hall, 1976.

[15] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.