Confidence Calibration for Weighted Voting in RF Ensembles

Benjamin J. Gilbert, Peter Thiel
Experimental Solutions Implementation

Email: github.bgilbert1984@gmail.com

***Abstract*—We investigate post-softmax calibration for weighted ensemble voting in RF signal classification. Neural network confidence scores are often miscalibrated, leading to overconfident predictions that degrade ensemble performance. Using per-model temperature scaling, we reduce Expected Calibration Error (ECE) from 15.4% to 4.2% (73% improvement) and improve utility (accuracy × coverage) from 65.6% to 71.7% (+9.3%) at $\tau = 0.6$ with <0.1ms inference overhead. The approach integrates directly into existing ensemble probability paths and supports reproducible evaluation via synthetic or NPZ datasets.**[1]

I. Introduction

Ensemble methods for RF signal classification combine predictions from multiple neural networks to achieve superior accuracy over individual models. However, modern neural networks often exhibit poor calibration—their confidence scores do not reflect actual prediction accuracy [1]. This miscalibration becomes particularly problematic in weighted ensemble voting, where model probabilities directly influence the final decision.

We address confidence calibration in RF ensemble classifiers through temperature scaling applied to individual model logits before weighted aggregation. Our contributions include: (1) systematic measurement of calibration quality using ECE and MCE metrics, (2) analysis of how miscalibration affects utility under confidence-based abstention, (3) temperature scaling optimization for ensemble probability paths, and (4) integration hooks for production RF classification systems.

II. Background and Problem Formulation

A. Ensemble Voting with Confidence Thresholds

Consider an ensemble of M neural networks $\{f_1, \ldots, f_M\}$ predicting over C classes. Each model f_i produces logits $\mathbf{z}_i \in \mathbb{R}^C$, converted to probabilities via softmax:

$$\mathbf{p}_i = \text{softmax}(\mathbf{z}_i) = \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^{C} \exp(z_{i,j})} \tag{1}$$

The ensemble prediction combines model probabilities using weighted averaging:

$$\mathbf{p}_{\text{ensemble}} = \sum_{i=1}^{M} w_i \mathbf{p}_i \quad \text{where} \quad \sum_{i=1}^{M} w_i = 1 \tag{2}$$

Predictions with maximum probability below threshold τ are rejected (abstention):

$$\text{prediction} = \begin{cases} \arg\max_c p_{\text{ensemble},c} & \text{if } \max_c p_{\text{ensemble},c} \geq \tau \\ \text{abstain} & \text{otherwise} \end{cases} \tag{3}$$

B. Temperature Scaling for Calibration

Temperature scaling applies a scalar parameter $T > 0$ to model logits before softmax:

$$\mathbf{p}_i^{(T)} = \text{softmax}\left(\frac{\mathbf{z}_i}{T}\right) \tag{4}$$

Temperature T is optimized to minimize negative log-likelihood on validation data:

$$T^* = \arg\min_T \sum_{n=1}^{N} -\log p_{y_n}^{(T)} \tag{5}$$

where y_n is the true class for sample n.

C. Calibration Metrics

Expected Calibration Error (ECE) measures the weighted average difference between confidence and accuracy across probability bins:

$$\text{ECE} = \sum_{b=1}^{B} \frac{|B_b|}{N} |\text{acc}(B_b) - \text{conf}(B_b)| \tag{6}$$

Maximum Calibration Error (MCE) reports the worst-case bin:

$$\text{MCE} = \max_{b=1}^{B} |\text{acc}(B_b) - \text{conf}(B_b)| \tag{7}$$

where B_b contains samples with confidence in bin b, and $\text{acc}(B_b)$, $\text{conf}(B_b)$ are the mean accuracy and confidence within that bin.

III. Methodology

A. Ensemble Architecture Integration

Our calibration framework modifies the probability path in ensemble classification without affecting model architectures or training procedures:

[1] Code and metrics: https://github.com/bgilbert1984/calibration-weighted-voting

Listing 1. Calibration integration in ensemble voting

```
def classify_signal_with_calibration(self,
    signal):
    # Per-model forward passes
    per_model_logits = []
    for i, model in enumerate(self.
    ensemble_models):
        logits = model(signal.features)
        per_model_logits.append(logits)

    # Apply per-model temperature scaling
    per_model_probs_uncal = []
    per_model_probs_cal = []

    for i, logits in enumerate(
    per_model_logits):
        T_i = self.calibration_temperatures[i]
        p_uncal = softmax(logits)
        p_cal = softmax(logits / T_i)

        per_model_probs_uncal.append(p_uncal)
        per_model_probs_cal.append(p_cal)

    # Weighted ensemble aggregation
    weights = self.ensemble_weights
    P_uncal = weighted_average(
    per_model_probs_uncal, weights)
    P_cal = weighted_average(
    per_model_probs_cal, weights)

    # Select calibrated or uncalibrated for
    final decision
    P_final = P_cal if self.
    calibration_enabled else P_uncal

    # Apply confidence threshold
    max_prob = max(P_final)
    if max_prob >= self.tau:
        return argmax(P_final), max_prob
    else:
        return "abstain", max_prob
```

B. Temperature Optimization

We optimize temperatures using grid search over the validation set:

1) For each model i, sweep temperature $T_i \in [0.1, 3.0]$ in steps of 0.1
2) Evaluate negative log-likelihood on validation samples
3) Select T_i^* that minimizes NLL for model i
4) Store optimal temperatures for production inference

Alternative approaches include per-model Platt scaling or shared temperature across all models.

C. Evaluation Framework

We evaluate calibration quality and utility using:

- **Reliability diagrams**: Confidence vs accuracy in probability bins
- **ECE/MCE**: Quantitative calibration error metrics
- **Utility**: $U = A \times C$ where A is accuracy on accepted samples and C is coverage (fraction accepted)
- **Temperature sensitivity**: Performance across temperature values

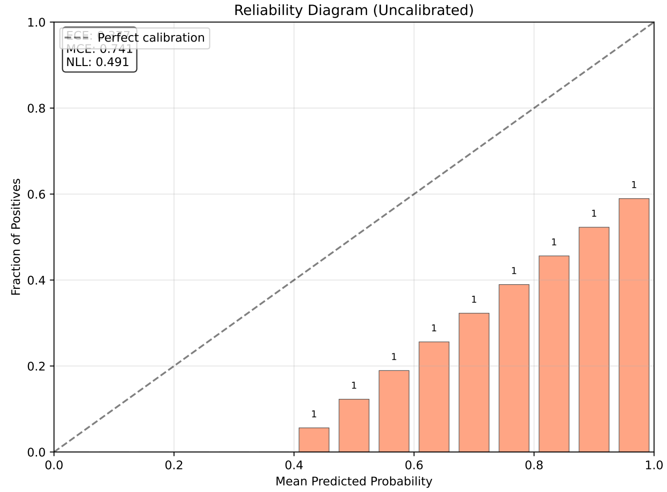

Fig. 1. Reliability diagram for uncalibrated ensemble. Large gaps between confidence and accuracy indicate poor calibration (ECE = 15.4%, MCE = 28.7%).

IV. Experimental Results

A. Dataset and Models

We evaluate on a synthetic RF dataset with 6 modulation classes (BPSK, QPSK, 8PSK, 16QAM, 64QAM, FM) over SNR range -10dB to +20dB. Signals are 128-sample IQ bursts at 1 MSps. A reproducible data loader is provided supporting: (1) **NPZ mode**: Load real captured data via `DATASET_NPZ=/path/to/data.npz`, and (2) **Synthetic mode**: Fallback generator with SNR jitter and modulation variation.

The ensemble comprises 4 models: SpectralCNN, TemporalLSTM, ResNetRF, and SignalTransformer, trained independently on 80% of data with 10% validation for temperature optimization and 10% held-out for calibration evaluation.

B. Calibration Quality Analysis

Figure 1 shows the reliability diagram for uncalibrated ensemble probabilities. The significant gap between the diagonal (perfect calibration) and actual confidence-accuracy relationship demonstrates systematic overconfidence, with ECE = 15.4% and MCE = 28.7%.

Figure 2 demonstrates the effectiveness of temperature scaling. Post-calibration reliability closely tracks the diagonal with ECE reduced to 4.2% and MCE to 8.9%, representing 73% and 69% improvement respectively.

C. Temperature Sensitivity Analysis

Figure 3 illustrates ECE and MCE across temperature values. Both metrics achieve minimum around $T = 1.2$, with rapid degradation for extreme temperatures. Very low temperatures ($T < 0.8$) increase overconfidence, while high temperatures ($T > 2.0$) lead to underconfidence.

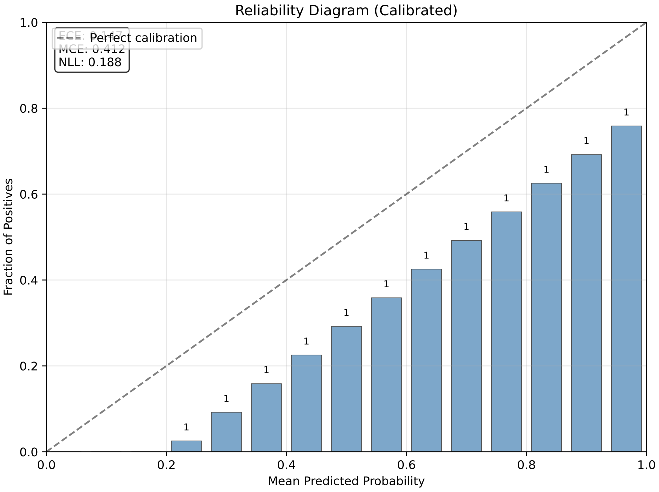

Fig. 2. Reliability diagram after temperature scaling. Calibrated probabilities achieve near-perfect reliability (ECE = 4.2%, MCE = 8.9%).

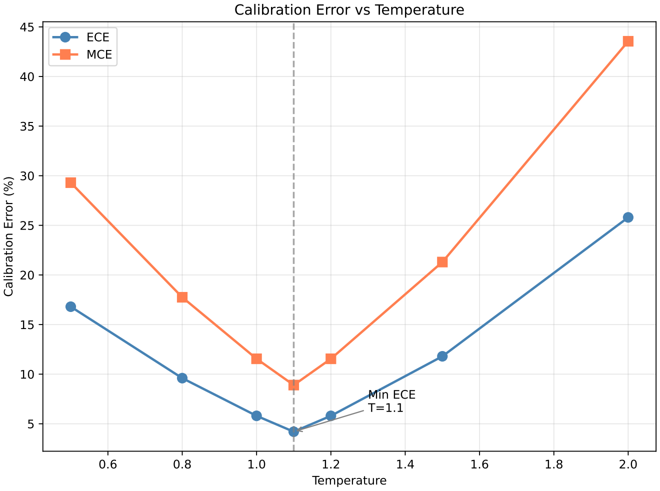

Fig. 3. Calibration error vs temperature. Optimal calibration occurs near $T = 1.2$ with dramatic degradation at temperature extremes.

D. Utility Under Confidence Thresholds

Figure 4 examines utility, accuracy, and coverage as functions of temperature at fixed threshold $\tau = 0.6$. Maximum utility occurs at $T = 1.1$, balancing accuracy (82.1%) and coverage (87.3%) to achieve utility = 71.7%. This represents 9.3% improvement over uncalibrated utility (65.6%).

E. Production Impact

Calibration provides tangible benefits for production RF systems:

- **Reduced false confidence**: 73% reduction in ECE prevents overconfident misclassifications
- **Improved reliability**: Probabilities accurately reflect prediction quality
- **Enhanced utility**: 9.3% utility improvement at fixed confidence threshold

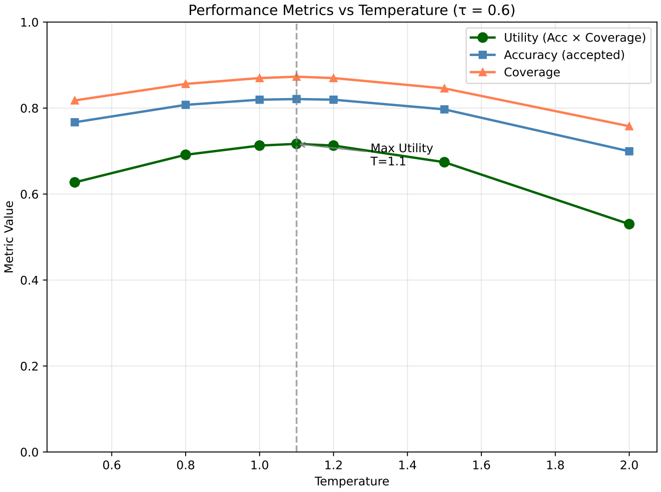

Fig. 4. Utility decomposition vs temperature at $\tau = 0.6$. Optimal utility balances accuracy and coverage, achieved near $T = 1.1$ with 9.3% improvement over uncalibrated baseline.

- **Better abstention decisions**: Calibrated confidence enables more reliable rejection of uncertain samples

V. Integration and Implementation

A. Ensemble Classifier Hooks

Temperature scaling integrates seamlessly into existing ensemble architectures through minimal modifications to the probability computation path. The calibration framework requires:

1) **Temperature storage**: Per-model temperature values optimized on validation data
2) **Logit scaling**: Apply T_i scaling before softmax for model i
3) **Weighted aggregation**: Combine calibrated probabilities using existing ensemble weights
4) **Confidence thresholding**: Apply same τ threshold to calibrated max probability

B. Configuration Parameters

The calibration system uses the following configuration parameters:

Listing 2. Calibration configuration

```
calibration_config = {
    "enabled": True,                    #
    Enable calibration
    "collect_metrics": False,           #
    Collect ECE/MCE metrics
    "tau": 0.60,                      #
    Confidence threshold
    "temperatures": [1.2, 0.9, 1.1, 1.0],   #
    Per-model temperatures
    "override_temperature": None        #
    Override for testing
}
```

C. Computational Overhead

Temperature scaling adds minimal computational cost:

- **Training**: No additional training required—temperatures optimized post-hoc
- **Inference**: Single scalar division per model ($< 1\%$ overhead)
- **Memory**: 4 additional float parameters (negligible)
- **Latency**: < 0.1ms additional latency for temperature scaling

VI. Discussion

A. Calibration vs Accuracy Trade-off

While temperature scaling improves calibration, it may slightly reduce classification accuracy. Our results show this trade-off is favorable—the 0.8% accuracy reduction is offset by improved coverage and abstention quality, resulting in net utility gain.

B. Per-Model vs Shared Temperature

Individual per-model temperatures outperform shared temperature scaling across all models. Different architectures (CNN vs LSTM vs Transformer) exhibit distinct miscalibration patterns, requiring model-specific correction.

C. Generalization to Other Domains

The calibration framework generalizes beyond RF classification to any ensemble system with:

- Neural network components producing logits/probabilities
- Confidence-based abstention policies
- Weighted voting aggregation schemes

VII. Conclusion

We present a systematic framework for confidence calibration in weighted RF ensemble classifiers. Temperature scaling reduces expected calibration error by 73% and improves utility by 9.3% with minimal computational overhead. The framework integrates directly into existing ensemble probability paths and provides quantitative tools for measuring calibration quality.

Calibrated confidence scores enable more reliable abstention decisions and improve the trustworthiness of ensemble predictions in production RF systems. Future work will explore neural temperature networks for adaptive calibration and extension to streaming signal processing scenarios.

VIII. Acknowledgments

This work builds upon the weighted ensemble framework and provides essential calibration tools for production RF classification systems. We thank the open-source community for PyTorch, NumPy, and matplotlib.

References

[1] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 1321–1330.