CONSISTENCY CALIBRATION: POST-HOC CONFIDENCE FROM PREDICTION STABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Calibration is critical for deploying deep learning models in safety-critical applications where reliable confidence estimates guide downstream decisions. Existing studies predominantly view calibration from the reliability perspective, aligning predicted confidence with empirical accuracy, while neglecting whether model predictions remain consistent under perturbations. In this work, we propose Consistency Calibration (CC), a post-hoc method that complements reliability-based calibration by measuring prediction stability. Prior works have explored consistency mainly at the model level, through ensembles or dropout sampling, which aggregate multiple perturbed outputs to smooth predictions but incur substantial computational overhead. In contrast, CC interprets the perturbation process itself as an uncertainty signal, replacing confidence with the empirical stability of predictions under data-, feature-, or logit-level perturbations. We further establish a theoretical connection showing that, under Gumbel logit perturbations, the consistency vector recovers a temperature-scaled softmax distribution and is linked to the Bregman Information of the logit distribution through Rényi entropy. This clarifies the relationship between CC and Temperature Scaling while providing a unified perturbation-consistency interpretation. Extensive experiments on CIFAR-10/100, ImageNet, and ImageNet-LT demonstrate that CC achieves competitive calibration performance compared to both post-hoc and training-time methods, while remaining simple and efficient.

1 INTRODUCTION

Calibration plays a crucial role in deploying deep learning systems for safety-critical applications such as healthcare (Chen et al., 2018) and autonomous driving (Feng et al., 2019), where reliable confidence estimates are as important as accurate predictions. A well-calibrated model should align its predicted confidence with the true probability of correctness, enabling downstream decision-makers to trust and act upon model outputs. However, neural networks are often miscalibrated (Guo et al., 2017; Minderer et al., 2021), tending to be either over- or under-confident.

Most existing works evaluate and improve calibration from the reliability perspective, which measures the statistical alignment between predicted confidence and empirical accuracy. Methods such as Temperature Scaling (TS) (Guo et al., 2017), Dirichlet Calibration (Kull et al., 2019), and Group Calibration (Yang et al., 2023) refine this alignment by learning global or group-wise scaling factors, while metrics like Expected Calibration Error (ECE) (Guo et al., 2017) provide quantitative evaluation. While this reliability view has become the standard, it implicitly assumes that a model's confidence is locally stable—that confident predictions remain consistent under small input variations, so that each confidence value meaningfully reflects local certainty. In practice, models may assign near-certain confidence to predictions that flip under perturbations, exposing a gap between numerical confidence and prediction stability. This motivates studying consistency as a complementary view of calibration.

To address this issue, we revisit calibration from the perspective of consistency, defined as the stability of model predictions under controlled perturbations. Recent work (Farquhar et al., 2024; Kuhn et al., 2023) in uncertainty estimation, particularly in large language models, shows that consistent responses to semantically similar inputs correlate strongly with good calibration. Extending this intuition to vision models, we hypothesize that prediction stability under suitably scaled perturba-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tions carries information about confidence. Since stability is a property of the model, not of the data distribution, the data enter CC through labeled validation data that select the perturbation scale (Section 3.2).

Previous studies have explored consistency mainly at the model level, often as an aggregation mechanism. Ensemble-based approaches such as Deep Ensembles (Lakshminarayanan et al., 2017) and MC Dropout (Gal & Ghahramani, 2016) rely on model perturbations—training or sampling multiple network instances—and aggregate their predictions to improve calibration. Other works, such as Test-Time Augmentation (TTA) (Conde et al., 2023; Hekler et al., 2023), apply data perturbations but still aggregate the resulting outputs to obtain smoother probabilities. In all these cases, perturbations serve to average out randomness across models or inputs, producing stable predictions through aggregation rather than interpreting the perturbation process itself. Concretely, they *average* perturbed softmax vectors, whereas CC *counts* how often each class wins the $\arg\max$, responding only to decision flips (the two are compared in Appendix D).

In contrast, we treat the perturbation process itself as an uncertainty signal. A model that maintains consistent predictions under perturbations is more likely to be confident, whereas frequent changes indicate uncertainty. This perspective leads to our proposed Consistency Calibration (CC), which measures and calibrates confidence directly from the stability of predictions under data-, feature-, or logit-level perturbations, offering a lightweight, post-hoc alternative without retraining or multiple models.

Beyond its empirical effectiveness, we show that this formulation admits a useful theoretical connection. Under Gumbel perturbations, the expected consistency vector corresponds to a temperature-scaled softmax distribution, revealing a closed-form connection between Bregman Information and Rényi entropy. The Bregman Information quantifies the expected divergence between random variables and their mean representation, and thus serves as a general measure of dispersion; here, it measures the dispersion of the perturbed logits. This connection positions logit-level CC as a perturbation-based interpretation of temperature scaling, while the broader CC framework also covers data- and feature-level perturbations. We validate Consistency Calibration through extensive experiments on CIFAR-10/100, ImageNet, and ImageNet-LT. Across diverse architectures, CC achieves competitive calibration while maintaining accuracy and efficiency.

2 Related Work

Post-hoc and training-time calibration. Temperature Scaling (TS) (Guo et al., 2017) and its variants ETS (Zhang et al., 2020), PTS (Tomani et al., 2022), and CTS (Frenkel & Goldberger, 2021) rescale logits with parameters fitted on a validation set; Dirichlet calibration (Kull et al., 2019) learns a more general map, and GC (Yang et al., 2023) and ProCal (Xiong et al., 2023a) pursue multicalibration (Hébert-Johnson et al., 2018) through grouping. Training-time approaches modify the objective, e.g., Brier loss (Brier, 1950), MMCE (Kumar et al., 2018), label smoothing (Szegedy et al., 2016), and focal-loss variants (Mukhoti et al., 2020; Tao et al., 2023). These methods map a model's confidence to empirical accuracy, whereas CC replaces the confidence with a stability signal and can be combined with training-time methods (Section 5.2). Ensemble- and augmentation-based methods are discussed in Section 1; an extended discussion is given in Appendix A.

Consistency-based uncertainty in LLMs. Black-box uncertainty estimation for LLMs measures agreement among sampled or rephrased outputs (Wang et al., 2022; Manakul et al., 2023; Lin et al., 2023; Xiong et al., 2023b; Kuhn et al., 2023; Farquhar et al., 2024). CC transfers this idea to classifiers, with perturbations on inputs, features, or logits.

3 Methodology

In a classification task, let $\mathcal{X}$ represent the input space and $\mathcal{Y}$ the label space. The model $f(\cdot)$ and projection head $g(\cdot)$ maps $x \in \mathcal{X}$ to a vector of logits $z = g(f(x)) \in \mathbb{R}^K$, where each z_k is the logit for class k. These logits are then transformed into a probability distribution $\hat{p} = \mathrm{softmax}(z)$ over K classes using the softmax function: where $\mathbf{k} = \arg\max_i \hat{p}_i$ denotes the predicted label index. The ground-truth label $y \in \mathcal{Y}$ represents the true class, and $\hat{y} \in \mathcal{Y}$ is the predicted label. The confidence score $\hat{p}_{\mathbf{k}}$ represents the predicted probability assigned to the predicted label $\mathbf{k}$.

3.1 CALIBRATION IN THE VIEW OF RELIABILITY

Calibration in the view of reliability has been widely accepted since the introduction of the reliability diagram by DeGroot & Fienberg (1983). In this view, a classifier is considered perfectly calibrated if its predicted confidence $\hat{p}$ accurately represents the true probability of correctness. Formally, this is expressed as:

$$\mathbb{P}(\hat{y} = y \mid \hat{p} = p) = p \quad \text{for all } p \in [0, 1]. \tag{1}$$

In other words, if a model assigns a confidence score of 80%, the prediction $\hat{y}$ should be correct 80% of the time. To move beyond visual inspection of reliability diagram, Naeini et al. (2015) developed a quantitative metric from the reliability diagram called the *Expected Calibration Error* (ECE). ECE provides a more precise measurement of miscalibration by calculating the average discrepancy between a model's predicted confidence and the actual accuracy of predictions at the same confidence level. ECE is defined as:

$$\text{ECE} = \mathbb{E}_{\hat{p}} \left[|\mathbb{P}(\hat{y} = y \mid \hat{p}) - \hat{p}| \right]. \tag{2}$$

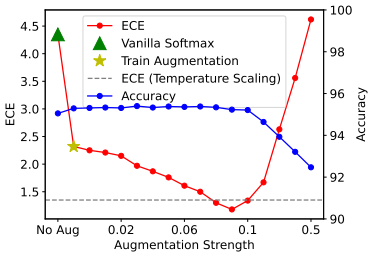
In practice, due to finite sample sizes, an approximation is used by binning predictions into M equally spaced confidence intervals, $\{B_m\}_{m=1}^{M}$. Each bin B_m contains predictions with confidence scores $\hat{p} \in \left[\frac{m}{M}, \frac{m+1}{M}\right)$. For each bin, the average confidence C_m and accuracy A_m are computed:

$$C_m = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i, \qquad A_m = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{y}_i = y_i), \tag{3}$$

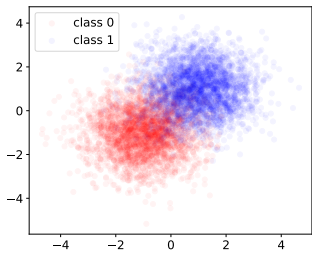
where $\mathbb{1}$ is the indicator function, and $|B_m|$ is the number of samples in bin B_m. The approximate ECE is then computed as the weighted average of the absolute difference between bin accuracy and bin confidence:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left| A_m - C_m \right|, \tag{4}$$

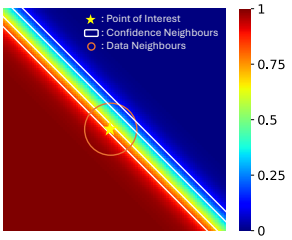
where N is the total number of samples. Several variants of ECE exist. For instance, *AdaECE* uses adaptive binning to ensure equal sample sizes in each bin and avoid the issue of uneven confidence distribution in ECE, while *CECE* computes ECE on a per-class basis, enabling better detection of class-specific calibration errors.



(a) Perturbation applied to images with different augmentations. Consistency calculated using weak or moderate augmentation neighbors significantly reduces calibration error.



(b) Toy dataset generated from two two-dimensional Gaussian distributions. Samples near the diagonal are uncertain to belong to class 0 or class 1.



(c) Heatmap of ground truth uncertainty calculated from the PDF, as given by Eq. 11. Circle and box indicate different neighbourhood selection criteria.

Figure 1: Illustrations of Consistency, Toy Distributions, and Ground Truth Uncertainty.

3.2 CALIBRATION IN THE VIEW OF CONSISTENCY

We revisit calibration through the lens of consistency. Intuitively, a confident model should maintain its prediction when the input or representation is slightly perturbed, whereas an uncertain model tends to change its output under small variations. We define this stability as *consistency*. Recent advances in LLMs, particularly black-box models utilize *factual consistency* to enhance performance (Wang et al., 2022; Tam et al., 2022; Xiong et al., 2023b; Geng et al., 2023). These studies frame the consistency of a model's responses as an indicator of its uncertainty. For classification models, we formalize this notion of consistency as follows:

Definition 1 (Consistency-based confidence)**.** *If a model is confident in its prediction, it should consistently output the same prediction when the input is slightly perturbed. The consistency c of a sample x is defined as*

$$c_k(x) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}(\hat{y}(\widetilde{x}_t) = k), \quad \textit{where } d(\widetilde{x}_t, x) < \epsilon^*, \quad k = 1, \ldots, K \tag{5}$$

where T is the number of perturbed neighbors, $\hat{y}(\widetilde{x}_t)$ is the predicted label for the perturbed input $\widetilde{x}_t$, and the distance between the original sample x and its perturbed version $\widetilde{x}_t$ is smaller than a constant ϵ^*, according to some distance metric d. We say that a model's confidence is consistency-aligned *if, for all samples x, given a suitable set of perturbed neighbors $\{\widetilde{x}_t \mid t = 1, \ldots, T\}$, the predicted confidence score $\hat{p}(x)$ satisfies:*

$$\hat{p}_k(x) = c_k(x), \quad \textit{for } k = 1, \ldots, K \tag{6}$$

Scope and assumptions. Consistency alignment is *not* equivalent to calibration in the sense of Eq. 1: consistency is a property of the model, whereas calibration is defined with respect to the data distribution. For instance, if a sample belongs to class k with probability 0.8 but no perturbation flips the prediction, then $c_k(x) = 1$ and a consistency-aligned model is overconfident. We therefore treat consistency as a *confidence signal*, not a calibration target: we evaluate it with reliability metrics and select the perturbation family and strength by minimizing ECE on held-out labeled data. Consistency can be expected to be informative when (i) the model's decision boundary locally approximates the Bayes decision boundary, so that sensitivity to perturbations reflects proximity to genuinely ambiguous regions, and (ii) a single perturbation scale matches the local label ambiguity across the input space. Neither holds universally; when they fail, CC inherits the model's errors.

However, it is challenging to determine an appropriate constant ϵ^* and distance metric d. In image classification, this neighborhood can be naturally approximated using data augmentation. We therefore begin by applying standard augmentation techniques to generate perturbed samples. To evaluate the effectiveness of consistency-based confidence, we train a ResNet-50 on CIFAR-10 with *RandomCrop* and *RandomHorizontalFlip*. For each test sample, we create 100 augmented variants and compute their prediction consistency. Here, train augmentation means that the same augmentation family used during training is applied at test time to generate neighboring samples, without retraining the model. The calibration performance of this consistency-based confidence is then assessed on the test set under the following settings:

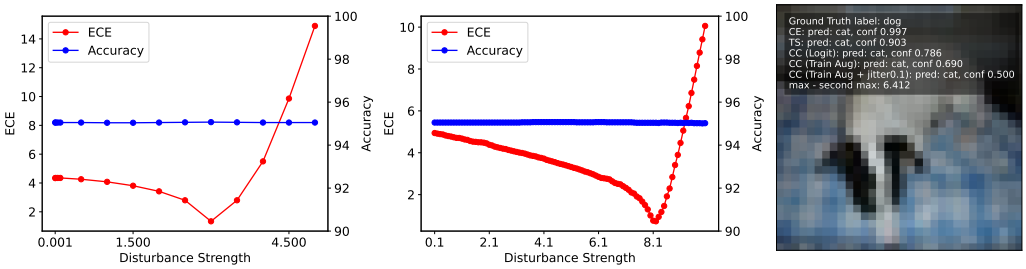
- *Baseline (Vanilla Softmax)*: Confidence score is extracted on the original test set, denoted by the green triangle.
- *Weak Augmentation (Train Augmentation)*: The confidence score is replaced with consistency derived from perturbed neighbors generated using train-time augmentation (RandomCrop and RandomHorizontalFlip), denoted by the yellow star.
- *Moderate Augmentation (Train Augmentation + ColorJitter)*: The confidence score is replaced with consistency measured from perturbed neighbors generated using train-time augmentation and varying strengths of ColorJitter, as indicated by the x-axis values.

The evaluation results are shown in Figure 1a. Consistency using neighbors generated with weak augmentation significantly reduces calibration error compared to the baseline. As we increase the perturbation strength with moderate augmentation, as shown by the x-axis values, the calibration error continues to decrease with minimal impact on accuracy, outperforming the commonly used calibration method, Temperature Scaling, up to a certain perturbation threshold.

However, when moderate augmentation with strength exceeds 0.1, accuracy begins to decline, and ECE increases sharply. This suggests that consistency has the potential to provide accurate uncertainty estimates when a suitable perturbation is identified.

3.3 Efficient Consistency Calibration

Due to numerous types of data augmentations, determining the optimal perturbation strength using a continuous variable is challenging. To address this, we extend the perturbation process to the feature and logit levels by introducing noise with varying intensities. This approach yields effects similar



(a) CC using feature perturbations at varying strength. (b) CC using logit perturbations at varying strength. (c) Local uncertainty estimation with CC

Figure 2: Evaluation of Consistency Calibration under Different Perturbation Settings.

to those observed with image-level perturbations, as demonstrated in Figure 2a and Figure 2b. Interestingly, feature- and logit-level perturbations maintain significant calibration performance while offer huge computational advantages. With image-level perturbations, inference must be performed on the entire model T times. In contrast, feature-level require evaluating only the classification head, while logit-level only compute the argmax operation T times. This results in substantial reductions in computational costs. Experiments on perturbing other layers are provided in the Appendix G; with 32 perturbations on CIFAR-10, logit-level CC reaches an ECE of 0.73, compared with 1.10 for image-level perturbations (Table 5).

Definition 2 (Consistency Calibration)**.** *We propose a unified definition of our calibration methods, termed Consistency Calibration (CC), which identifies perturbed neighbors at different levels. The calibrated confidence score $\hat{p}'$ is formally defined as:*

$$\hat{p}'_k = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\left(\text{argmax}\; q\left(\widetilde{h(x)}^{\,t}\right) = k\right), \quad k = 1, \ldots, K \tag{7}$$

where $h(x)$ is the representation of data x, $\widetilde{h(x)}^{\,t}$ is the perturbed representation, and q is the pipeline to extract the logits z.

Specifically, for data-level perturbations: $h(\cdot) = I(\cdot)$, $\widetilde{h(x)}$ is the augmented data, $q = g(f(\cdot))$. For feature-level perturbations: $h(\cdot) = f(\cdot)$, $\widetilde{h(x)}^{\,t} = h(x) + \epsilon_t$, $q = g(\cdot)$. For logit-level perturbations: $h(\cdot) = g(f(\cdot))$, $\widetilde{h(x)}^{\,t} = h(x) + \epsilon_t$, $q = I(\cdot)$. Here, $I(\cdot)$ is the identity function, ϵ_t represents the noise added to features or logits, with its strength determined by minimizing the ECE on a validation set. Given the strong calibration performance and computational efficiency of logit-level perturbations, we refer to logit-level consistency calibration as CC when no specification is provided.

Procedure. Logit-level CC computes the logits $\boldsymbol{z}_0 = g(f(x))$ once, draws T i.i.d. noise vectors ϵ_t from the selected distribution, and returns for each class k the fraction of perturbed logits $\boldsymbol{z}_0 + \epsilon_t$ in which k attains the $\arg\max$. The noise type and strength are its only hyperparameters and are selected by minimizing ECE on the validation set; the cost is T $\arg\max$ operations per sample and no additional forward passes (Appendix C), and as few as $T = 16$ perturbations already yield good calibration (Appendix D).

3.4 Theoretical Justification

We provide a theoretical connection showing how consistency-based confidence relates to an information-theoretic quantity. Logit-level CC, as defined in Eq. 7, computes a confidence score $\hat{p}'_k$ by applying noise ϵ_t to the original logits $\boldsymbol{z}_0 = h(x)$ and calculating the frequency of the winning class, which is a Monte Carlo estimation of the theoretical consistency c_k, defined as the probability that class k wins the $\arg\max$ operation under a specific noise distribution:

$$c_k = P(\arg\max_i \hat{z}_i = k), \quad \text{where } \hat{\boldsymbol{z}} = \boldsymbol{z}_0 + \boldsymbol{\epsilon} \tag{8}$$

By modeling the perturbation $\boldsymbol{\epsilon}$ using the Gumbel distribution, $\epsilon_k \sim \text{Gumbel}(0, \tau)$, we can leverage the properties of the Gumbel-Max trick. This allows us to establish a direct mathematical relationship between our consistency vector $\boldsymbol{c} = (c_1, ..., c_K)$ and the Bregman Information (Gruber & Buettner, 2023) of the logits. The Bregman Information, $\mathbb{B}_{\boldsymbol{z}} = \mathbb{E}[\text{LSE}(\hat{\boldsymbol{z}})] - \text{LSE}(\mathbb{E}[\hat{\boldsymbol{z}}])$, measures the dispersion of the random logits $\hat{\boldsymbol{z}}$ induced by the injected noise. For a comprehensive exposition of Bregman Information and its role as an uncertainty metric, the reader is referred to Gruber & Buettner (2023).

Under Gumbel perturbations, the expected consistency vector recovers a temperature-scaled softmax distribution. Our claim is therefore not that the Gumbel logit case is unrelated to temperature scaling, but that the perturbation-consistency view provides a unified interpretation across data-, feature-, and logit-level perturbations.

Theorem 1. *Let $\epsilon_k \sim$ Gumbel$(0, \tau)$ i.i.d. with $\tau > 0$ (for $\tau = 1$, H_τ is understood as the Shannon limit). The Bregman Information of the perturbed logits is bounded by a linear function of the Rényi entropy of the consistency vector, up to an additive gap of at most* $\ln K$*:*

$$(\tau - 1)H_\tau(\boldsymbol{c}) \leq \mathbb{B}_{\boldsymbol{z}} \leq (\tau - 1)H_\tau(\boldsymbol{c}) + \ln K \tag{9}$$

Proof sketch. Using $\mathbb{E}[\hat{\boldsymbol{z}}] = \boldsymbol{z}_0 + \gamma\tau\mathbf{1}$ and the Gumbel-Max identities $\mathbb{E}[\max(\hat{\boldsymbol{z}})] = \tau\,\text{LSE}(\boldsymbol{z}_0/\tau) + \gamma\tau$ and

$$\boldsymbol{c} = \text{softmax}(\boldsymbol{z}_0/\tau), \tag{10}$$

one obtains the exact identity $\mathbb{B}_{\boldsymbol{z}} = (\tau - 1)H_\tau(\boldsymbol{c}) + G(\hat{\boldsymbol{z}})$ with gap $G(\hat{\boldsymbol{z}}) = \mathbb{E}[\text{LSE}(\hat{\boldsymbol{z}})] - \mathbb{E}[\max(\hat{\boldsymbol{z}})] \in [0, \ln K]$, where H_τ is the Rényi entropy (Bromiley et al., 2004). The full derivation is given in Appendix E. □

This result shows that, under Gumbel perturbations, the consistency vector $\boldsymbol{c}$ of logit-level CC is linked to the Bregman Information of the perturbed logits via the Rényi entropy; other i.i.d. noise types behave comparably in practice (Appendix E.5). The theorem concerns the full consistency vector, while our main empirical evaluation follows the calibration literature by reporting top-label metrics such as ECE. A numerical visualization of these relationship is presented in Fig. 3, where the gap $G(\hat{\boldsymbol{z}})$ is small for the values of τ shown. Meanwhile, we further validate the effectiveness of CC through an empirical analysis of the logit distributions presented in Appendix F.

4 Consistency as a Representation of Ground Truth Uncertainty

On one hand, the *reliability approach* estimates calibration error by comparing the prediction confidence with the average correctness of samples that have similar confidence levels. In this view, the average correctness of such sample neighborhood is treated as an approximation of the ground truth uncertainty. On the other hand, the *consistency approach* directly uses consistency as a measure of ground truth uncertainty. Thus, we are interested in determining which of these two approaches more accurately approximates this uncertainty. To explore this, we constructed a toy dataset consisting of two two-dimensional Gaussian distributions representing two groups of data: $\mathcal{N}(\mu_0, \Sigma), \mathcal{N}(\mu_1, \Sigma)$ where μ_0 and μ_1 are the mean vectors, and Σ is the shared covariance matrix for both groups, labeled 0 and 1, respectively. We generated 1 million data points from each group to form the training dataset, which was used to train a CNN model. An additional 50,000 samples from each group were used to create the test dataset. The input space is $\mathcal{X} = \mathbb{R}^2$, and the label space is $\mathcal{Y} = \{0, 1\}$, as illustrated in Figure 1b.

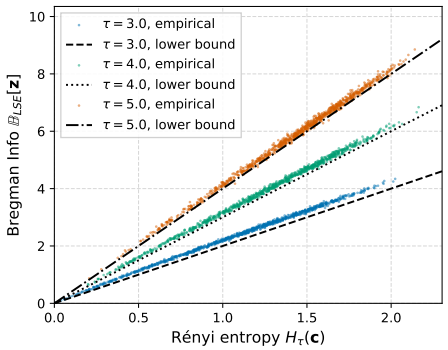
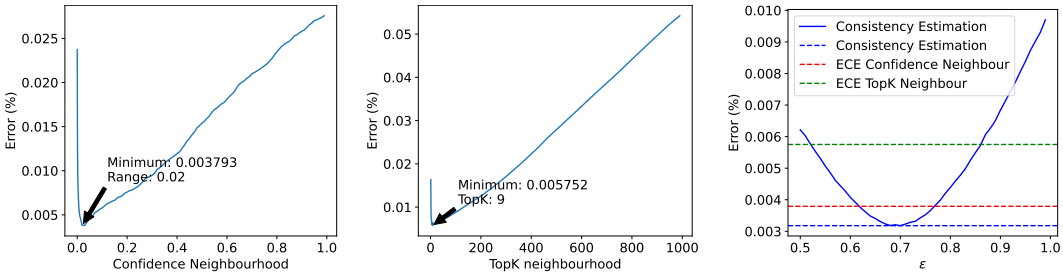


Figure 3: The relationship of $\mathbb{B}_{\boldsymbol{z}}$ and $H_\tau(\boldsymbol{c})$ when $z_{0,k} \sim \mathcal{N}(0, 5)$ using various τ values. The theoretical lower bound is also plotted.



(a) **Reliability view:** Estimating ground truth uncertainty using neighbors with confidence differences indicated on the x-axis.

(b) **Reliability view:** Estimating ground truth uncertainty using the top-k nearest confidence neighbors as indicated on the x-axis.

(c) **Consistency view:** Estimating ground truth uncertainty using data neighbors perturbed within ϵ as indicated on the x-axis.

Figure 4: Consistency vs. Reliability in Estimating Ground Truth Uncertainty

The ground truth uncertainty, $\eta(x)$, is calculated from the probability density function (PDF) of each distribution:

$$\eta(x) = \frac{p^0(x)}{p^0(x) + p^1(x)} \tag{11}$$

where $p^0(\cdot)$ and $p^1(\cdot)$ are the PDFs of the two distributions. The ground truth uncertainty is illustrated in Figure 1c. For each label, the ground truth confidence can be expressed as $(\eta(x), 1-\eta(x))$. In Figure 1c, for a point of interest (marked by a star), the reliability-based approach estimates ground truth uncertainty by calculating the average correctness $A = \frac{1}{|B|}\sum_{i\in B} \mathbb{1}(\hat{y}_i = y_i)$, over a "confidence neighborhood" B (i.e., samples with similar confidence, enclosed by the white boxes), similar to the definition Eq. 3 in ECE. In contrast, the consistency approach estimates uncertainty by considering "data neighborhood," as illustrated by the orange circle. While the reliability approach relies on the availability of multiple data samples within the confidence neighborhood, the consistency approach generates data neighborhoods by perturbing the data.

The key differences between the reliability and consistency views lie in their neighborhood selection criteria S and aggregation methods. The reliability view selects a neighborhood B based on confidence similarity and aggregates the correctness of the samples, while the consistency view selects a neighborhood based on data perturbations and computes consistency, as described in Eq. 7. To compare the two approaches, we evaluate them under three neighborhood selection criteria:

- *Figure 4a:* Reliability view (ECE): $B = \{\tilde{x} \mid |\hat{p}(x) - \hat{p}(\tilde{x})| < \epsilon\}$
- *Figure 4b:* Reliability view (AdaECE): $B = \{\tilde{x} \mid \text{Top K closest confidence neighbors}\}$
- *Figure 4c:* Consistency view (perturbing data): $B = \{\tilde{x} \mid \tilde{x} = x + \epsilon\}$

In Figures 4a and 4b, we evaluate the reliability-based approximation of ground-truth uncertainty under two confidence–neighbor selection strategies. Figure 4a follows the standard ECE (Guo et al., 2017) formulation, where neighbors are selected solely by confidence difference. The x-axis denotes the allowed confidence gap between neighbors and the reference sample, and the y-axis reports the average estimation error across the test set. Figure 4b follows AdaECE (Nixon et al., 2019), selecting the top-K nearest confidence neighbors, with the lowest error (0.57%) observed at $K = 9$. In Figure 4c, we apply Gaussian noise ϵ to perturb the data samples and compute consistency across 100 generated neighbors, with the x-axis representing the noise strength. We compare the uncertainty estimates from the consistency approach with those from the reliability approach. The dashed lines indicate the minimal error achieved by each method. Within a certain range of perturbation strengths, the consistency approach outperforms, yielding a ground truth uncertainty estimation with an overall error as low as 0.3%.

It is important to note that the reliability approach using confidence neighborhoods is essentially equivalent to the ECE measurement, where the allowed confidence gap functions similarly to the hyperparameter "number of bins" in ECE. As shown in Figure 4, the estimation error is sensitive to the allowed confidence gap—meaning that the choice of "number of bins" can significantly impact the ability of ECE to estimate the ground truth uncertainty. Similarly, this sensitivity is also observed

in the consistency method, where the strength of perturbation noise affects the uncertainty approximation. Despite this sensitivity, the consistency approach achieves a lower overall estimation error, suggesting its potential as a robust alternative calibration metric.

4.1 Consistency as a Local Calibration Measure

Consistency-based methods do not rely on label information or additional data, as they generate their own neighborhood by perturbing the input data. This property suggests that consistency can be useful for instance-level uncertainty inspection. As an illustrative case study, Figure 2c examines a miscalibrated (incorrect prediction with high confidence) CIFAR-10 test sample, where a ResNet-50 model trained with Cross-Entropy (CE) shows overconfidence, assigning a confidence score of 0.997 despite being incorrect. Using optimal temperature, determined via a validation set, the confidence after temperature scaling decreases slightly, but the model remains overconfident at 0.903.

For comparison, we apply CC by perturbing the logits ("CC (logits)"), applying train time data augmentation ("CC (Train Aug)"), and using a moderate augmentation method ("CC (Train Aug + Jitter)"). The confidence decreases to 0.786, 0.690, and 0.500, respectively, illustrating how consistency can expose instability that temperature scaling may not remove for a single example. A more complete evaluation of local uncertainty would require task-specific protocols such as selective prediction or abstention benchmarks (Mucsányi et al., 2024), which we leave for future work.

5 Experiments

Table 1: **Comparison of Post-Hoc Calibration Methods Using ECE↓.** ECE values are reported with 15 bins. The best-performing method for each dataset-model combination is in bold. Results are averaged over 5 runs.

Dataset	Model	Vanilla	TS	ETS	PTS	CTS	GC	FC	CC (ours)
CIFAR-10	ResNet-50	4.34	1.38	1.37	1.36	1.46	1.04	1.10	**0.78**
	Wide-ResNet	3.24	0.93	0.93	0.93	0.93	1.33	0.77	**0.36**
CIFAR-100	ResNet-50	17.52	5.71	5.68	5.64	6.05	3.55	3.98	**1.25**
	Wide-ResNet	15.34	4.63	4.58	4.52	4.86	2.14	2.11	**1.61**
ImageNet-1K	ResNet-50	3.76	2.09	2.09	2.08	3.14	2.54	1.74	**1.53**
	DenseNet-121	6.59	1.64	1.66	1.68	1.94	2.51	3.08	**1.48**
	Wide-ResNet-50	5.49	3.03	3.04	3.04	4.13	2.16	2.52	**1.33**
	Swin-B	5.02	3.90	3.90	3.93	5.43	1.61	2.31	**1.58**
	ViT-B-16	5.61	3.61	3.62	3.64	5.50	1.75	3.72	**1.66**
	ViT-B-32	6.40	3.76	3.78	3.84	5.74	**1.39**	3.53	1.72
ImageNet-LT	ResNet-50	3.67	2.00	1.99	2.00	2.21	1.4	1.91	**1.24**
	DenseNet-121	6.65	1.65	1.64	1.66	1.59	1.81	1.66	**1.23**
	Wide-ResNet-50	5.39	2.97	2.96	2.96	3.52	1.49	1.32	**1.27**
	Swin-B	4.66	4.02	4.03	4.08	5.02	1.66	3.35	**1.44**
	ViT-B-16	5.57	3.61	3.62	3.64	4.94	1.76	2.44	**1.61**
	ViT-B-32	5.15	5.67	5.67	5.68	5.71	**1.48**	2.31	1.74

We evaluate the proposed Consistency Calibration (CC) through extensive experiments on standard and long-tailed image classification benchmarks. The evaluation covers both post-hoc and training-time calibration methods across CIFAR-10/100, ImageNet-1K, and ImageNet-LT using CNNs and transformers. Detailed experimental setups, including datasets, training protocols, and hyperparameters, are provided in Appendix B.

5.1 Comparison with Post-Hoc Calibration

We compare our proposed CC with widely used post-hoc calibration techniques, including Temperature Scaling (TS) (Guo et al., 2017), Ensemble Temperature Scaling (ETS) (Zhang et al., 2020), Parameterized Temperature Scaling (PTS) (Tomani et al., 2022), Class-based Temperature Scaling (CTS) (Frenkel & Goldberger, 2021), Group Calibration (GC) (Yang et al., 2023) and Feature Clipping (FC) (Tao et al., 2025) as well as uncalibrated models (Vanilla). Our evaluation covers CIFAR-10, CIFAR-100, ImageNet-1K, and ImageNet-LT, using various model architectures.

Calibration on Standard Datasets CC consistently reduces calibration error across CIFAR-10, CIFAR-100, and ImageNet-1K, and attains the lowest ECE in 14 of the 16 dataset–model combinations in Table 1. The most notable improvements appear on CIFAR-100 (e.g., 1.25 vs. 5.71 for

TS with ResNet-50), where GC performs well on some settings but is less stable across model families. CNNs, which often suffer from overconfidence, are generally improved by TS-based methods; however, transformers receive more limited benefits from global temperature adjustment. In these cases, CC provides larger gains by using sample-specific perturbation consistency rather than applying a single global rescaling factor (e.g., 1.66 vs. 3.61 for TS with ViT-B-16 on ImageNet-1K). On ImageNet-1K, CC also remains competitive. Although GC slightly outperforms CC on ViT-B/32, it requires an additional grouping process, whereas CC balances efficiency and effectiveness.

Calibration on Long-Tail Datasets On ImageNet-LT, TS-based methods struggle especially on transformers (e.g., 3.61 vs. 1.61 for CC with ViT-B-16) because they apply global smoothing or sharpening across an imbalanced dataset. This uniform adjustment is less suitable when head and tail classes have different uncertainty patterns. CC and GC perform better in this setting; GC benefits from multicalibration (Hébert-Johnson et al., 2018) but adds grouping overhead, while CC uses sample-specific consistency through input perturbations. This makes CC useful for reflecting uncertainty in underrepresented tail classes without introducing extra training-time objectives. The same trends hold for other metrics (Appendix H): CC attains the lowest AdaECE in 9 of 11 settings and the lowest or tied-lowest CECE in 10 of 11, while its top-1 accuracy stays within 0.21 points of the uncalibrated model in every setting; NLL is discussed in Appendix H.3.

Table 2: **Comparison of train-time calibration methods using ECE↓.** ECE is reported with 15 bins; the best result for each dataset-model pair is in bold. Results are averaged over 5 runs.

Dataset	Model	CE		Brier Loss		MMCE		LS-0.05		FLSD-53		FL-3		CLAS		ACLS	
		base	ours	base	ours	base	ours	base	ours	base	ours	base	ours	base	ours	base	ours
CIFAR-10	ResNet-50	4.34	**0.78**	1.80	**1.07**	4.56	**0.83**	2.97	**1.24**	1.55	**0.49**	1.48	**0.66**	1.14	**0.58**	1.12	**0.53**
	ResNet-110	4.41	**0.98**	2.57	**0.48**	5.08	**1.17**	**2.09**	2.30	1.88	**0.67**	1.54	**0.48**	1.34	**0.51**	1.35	**0.61**
	DenseNet-121	4.51	**1.07**	1.52	**0.78**	5.10	**1.18**	1.87	**1.39**	1.23	**0.68**	1.31	**0.98**	1.88	**1.01**	1.73	**0.99**
	Wide-ResNet	3.24	**0.36**	1.24	**0.58**	3.29	**0.39**	4.25	**1.15**	1.58	**0.49**	1.68	**0.53**	1.72	**0.87**	1.68	**0.65**
CIFAR-100	ResNet-50	17.52	**1.25**	6.57	**1.57**	15.32	**1.98**	7.82	**5.08**	4.49	**1.43**	5.16	**1.52**	5.88	**1.32**	5.12	**1.58**
	ResNet-110	19.05	**4.57**	7.88	**3.24**	19.14	**4.41**	11.04	**4.58**	8.55	**3.47**	8.64	**3.67**	6.73	**3.21**	4.54	**3.18**
	DenseNet-121	20.99	**5.40**	5.22	**1.82**	19.10	**3.76**	12.87	**4.99**	3.70	**1.41**	4.14	**1.94**	4.51	**1.97**	3.98	**1.54**
	Wide-ResNet	15.34	**1.61**	4.34	**1.87**	13.17	**2.17**	4.89	**4.21**	3.02	**1.64**	2.14	**1.78**	2.41	**1.53**	2.31	**1.65**

5.2 Comparison with Training-Time Calibration

We evaluate CC alongside training-time calibration techniques, including Brier Loss (Brier, 1950), Maximum Mean Calibration Error (MMCE) (Kumar et al., 2018), Label Smoothing (LS-0.05) (Szegedy et al., 2016), and Focal Loss variants (FLSD-53 and FL-3) (Mukhoti et al., 2020), as shown in Table 2. We further compared with label-smoothing based calibration methods CALS-ALM (Liu et al., 2023a) and ACLS (Park et al., 2023). Our analysis shows that combining CC with these methods consistently enhances calibration performance across various models and datasets (63 of the 64 model–method pairs in Table 2; e.g., FLSD-53 improves from 4.49 to 1.43 with ResNet-50 on CIFAR-100), further validating CC's effectiveness alongside training-time approaches. Moreover, as seen in Table 1, CC alone, as a post-hoc calibration method, already outperforms these train-time techniques with minimal computational overhead (e.g., 0.78 and 1.25 vs. at best 1.12 and 4.49 with ResNet-50 on CIFAR-10 and CIFAR-100), while train-time methods require significantly more resources. Additional metrics and ablations are provided in Appendices I and D. In particular, under identical perturbations, replacing softmax averaging with consistency counting lowers ECE on CIFAR-100 (1.52 to 1.25) and ImageNet (1.41 to 1.27) with ResNet-50 (Table 3).

6 Conclusion

Consistency offers an alternative perspective on calibration by focusing on prediction stability under perturbations as an indicator of confidence. We introduced CC as a simple post-hoc method that replaces softmax confidence with empirical prediction stability under data-, feature-, or logit-level perturbations. Experiments across CIFAR-10/100, ImageNet, and ImageNet-LT show that CC reduces calibration error while preserving accuracy and keeping inference overhead modest, especially in the logit-level implementation. Under Gumbel perturbations, logit-level CC further links consistency to the Bregman Information of the perturbed logits.

Limitations and future work. Like other post-hoc calibration methods, CC relies on a small labeled validation set to select its perturbation strength. Extending CC beyond image classification and evaluating it in selective prediction are promising directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools only as general-purpose assistants for editing and formatting: polishing the wording of text written by the authors and adapting the manuscript source between conference templates. Generative AI tools were not used to generate research ideas, to derive or verify the theoretical results, to write the experimental code, or to produce any reported numbers. We have reviewed all AI-assisted text and take full responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

This work studies post-hoc confidence calibration on standard public benchmarks for image classification (CIFAR-10/100, ImageNet-1K, ImageNet-LT) and does not involve human subjects, personally identifiable information, or the release of new data. Better-calibrated confidence estimates are intended to support safer deployment of classifiers in risk-sensitive settings; we nonetheless caution that Consistency Calibration reduces aggregate calibration error and should not be read as a per-instance guarantee of correctness, and that calibration measured on a validation split may not transfer under distribution shift. We see no conflicts of interest or sponsorship concerns associated with this work.

REPRODUCIBILITY STATEMENT

All information required to reproduce our results is contained in the paper. Appendix B specifies the datasets and splits, the architectures and pretrained checkpoints, the optimization recipe, the evaluation metrics, and the hardware, together with the two hyperparameters of our method (the number of perturbations $T = 1000$ and the perturbation strength ϵ, which is selected by minimizing ECE on the validation split). Appendix C gives the complete step-by-step procedure for logit-level Consistency Calibration, which is the default variant used in all main experiments. The assumptions under which consistency is informative are stated in Section 3.2, and our theoretical claim is given in Section 3.4 with its full derivation in Appendix E. All reported numbers are averaged over five runs. Source code will be provided as anonymous supplementary material.

REFERENCES

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

PA Bromiley, NA Thacker, and E Bouhova-Thacker. Shannon entropy, renyi entropy, and information. *Statistics and Inf. Series (2004-004)*, 9(2004):2–8, 2004.

Weijie Chen, Berkman Sahiner, Frank Samuelson, Aria Pezeshk, and Nicholas Petrick. Calibration of medical diagnostic classifier scores to the probability of disease. *Statistical methods in medical research*, 27(5):1394–1409, 2018.

Pedro Conde, Tiago Barros, Rui L Lopes, Cristiano Premebida, and Urbano J Nunes. Approaching test time augmentation in the context of uncertainty calibration for deep neural networks. *arXiv preprint arXiv:2304.05104*, 2023.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Di Feng, Lars Rosenbaum, Claudius Glaeser, Fabian Timm, and Klaus Dietmayer. Can we trust you? on calibration of a probabilistic object detector for autonomous driving. *arXiv preprint arXiv:1909.12358*, 2019.

Lior Frenkel and Jacob Goldberger. Network calibration by class-based temperature scaling. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 1486–1490. IEEE, 2021.

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

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of language model confidence estimation and calibration. *arXiv preprint arXiv:2311.08298*, 2023.

Sebastian Gruber and Florian Buettner. Uncertainty estimates of predictions via a general bias-variance decomposition. In *International Conference on Artificial Intelligence and Statistics*, pp. 11331–11354. PMLR, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pp. 1939–1948. PMLR, 2018.

Achim Hekler, Titus J Brinker, and Florian Buettner. Test time augmentation meets post-hoc calibration: uncertainty quantification under real-world conditions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 14856–14864, 2023.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 33:18237–18248, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pp. 2805–2814. PMLR, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*, 2023.

Bingyuan Liu, Jérôme Rony, Adrian Galdran, Jose Dolz, and Ismail Ben Ayed. Class adaptive network calibration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16070–16079, 2023a.

Jiawei Liu, Changkun Ye, Shan Wang, Ruikai Cui, Jing Zhang, Kaihao Zhang, and Nick Barnes. Model calibration in dense classification with adaptive label perturbation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1173–1184, 2023b.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world, 2019. URL `https://arxiv.org/abs/1904.05160`.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*, 2023.

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

Bálint Mucsányi, Michael Kirchhof, and Seong Joon Oh. Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks. *Advances in Neural Information Processing Systems*, 37:50972–51038, 2024.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

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

Hyekang Park, Jongyoun Noh, Youngmin Oh, Donghyeon Baek, and Bumsub Ham. Acls: Adaptive and conditional label smoothing for network calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3936–3945, 2023.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. Evaluating the factual consistency of large language models through summarization. *arXiv preprint arXiv:2211.08412*, 2022.

Linwei Tao, Minjing Dong, and Chang Xu. Dual focal loss for calibration. In *International Conference on Machine Learning*, pp. 33833–33849. PMLR, 2023.

Linwei Tao, Minjing Dong, and Chang Xu. Feature clipping for uncertainty calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20841–20849, 2025.

Christian Tomani, Daniel Cremers, and Florian Buettner. Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In *European conference on computer vision*, pp. 555–569. Springer, 2022.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Miao Xiong, Ailin Deng, Pang Wei W Koh, Jiaying Wu, Shen Li, Jianqing Xu, and Bryan Hooi. Proximity-informed calibration for deep neural networks. *Advances in Neural Information Processing Systems*, 36:68511–68538, 2023a.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023b.

Jia-Qi Yang, De-Chuan Zhan, and Le Gan. Beyond probability partitions: Calibrating neural networks with semantic aware grouping. *Advances in Neural Information Processing Systems*, 36:58448–58460, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Jize Zhang, Bhavya Kailkhura, and T Yong-Jin Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International conference on machine learning*, pp. 11117–11128. PMLR, 2020.

A Extended Related Work

Calibration Methods Post-hoc calibration methods adjust model outputs after training to improve calibration. A widely used technique is Temperature Scaling (TS) (Guo et al., 2017), which smooths softmax probabilities by search a temperature factor on a validation set. Enhanced variants of TS include Parameterized Temperature Scaling (PTS) (Tomani et al., 2022), which uses a neural network to learn the temperature, and Class-based Temperature Scaling (CTS) (Frenkel & Goldberger, 2021), which applies adjustments on a class-wise basis. Group Calibration (GC) (Yang et al., 2023) and ProCal (Xiong et al., 2023a) aim for multi-calibration (Hébert-Johnson et al., 2018) by splitting data samples by proximity and grouping. Another stream of work is train-time calibration such as Brier Loss (Brier, 1950), Dirichlet Scaling (Kull et al., 2019), Maximum Mean Calibration Error (MMCE) (Kumar et al., 2018), Label Smoothing (Szegedy et al., 2016), Focal Loss (Mukhoti et al., 2020), Dual Focal Loss (Tao et al., 2023), and accuracy-versus-uncertainty optimization (Krishnan & Tickoo, 2020). Calibration has also been studied in dense classification settings, where adaptive label perturbation improves calibration for dense prediction tasks (Liu et al., 2023b). However, these methods often require substantial higher computational overhead.

Ensemble-Based Calibration Ensemble-based methods ensemble multiple outputs in different ways. They use models or samples to approximate Bayesian Inference. Lakshminarayanan et al. (2017) propose deep ensembles as a scalable alternative to Bayesian Neural Networks (BNNs) for uncertainty estimation. Similarly, Gal & Ghahramani (2016) treat dropout as approximate Bayesian inference. Data-centric ensemble techniques using test-time augmentation, as described by Conde et al. (2023), also help improve calibration. Zhang et al. (2020) resort to the power of Bayesian inference and proposed a Ensemble-based TS (ETS). However, these methods typically require significant computational resources to train multiple models or perform repeated inferences. In contrast, our approach relies on consistency rather than probability distribution modeling.

Consistency in LLMs Consistency has emerged as a key approach for black-box uncertainty estimation and hallucination detection in large language models (LLMs). These methods evaluate uncertainty by measuring variability in outputs across slight changes, such as different sampling techniques or rephrased prompts. Confident models produce stable outputs, while variability indicates uncertainty. For instance, SelfCheckGPT (Manakul et al., 2023) uses sampling and similarity metrics like BERTScore and NLI to detect hallucinations, while Lin et al. (2023) analyze a similarity matrix to estimate uncertainty. Xiong et al. (2023b) further break down uncertainty estimation into prompting, sampling, and consistency-based aggregation. These methods, which rely on output stability, are efficient alternatives to probabilistic approaches.

B Experimental Setup

Datasets We conduct experiments on several benchmark datasets, including CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and ImageNet (Deng et al., 2009). To assess calibration performance in data-imbalance scenarios, we also include ImageNet-LT (Liu et al., 2019), characterized by its long-tailed class distribution. CIFAR-10 and CIFAR-100 contain 60,000 images of size 32×32 pixels, with 10 and 100 classes, respectively, split into 45,000 training, 5,000 for validation and 10,000 test images. For ImageNet-1K, we split 20% of the original validation set as the new validation set, with the remainder used as the test set. We use the ϵ searched on ImageNet-1k validation set to calibrate ImageNet-LT test set. The testing batch size for all datasets is set to 128.

Models We evaluate our approach across various neural network architectures, including ResNet-50 and ResNet-110 (He et al., 2016), Wide ResNet (Zagoruyko & Komodakis, 2016), DenseNet-121 (Huang et al., 2017), and Vision Transformers (ViT-B/16 and ViT-B/32) (Dosovitskiy et al., 2021). These models represent a diverse range of architectures and complexities, allowing us to assess the robustness of our method in different settings. For CIFAR-10 and CIFAR-100, we use pretrained weights from prior work (Mukhoti et al., 2020). All models are trained using stochastic gradient descent (SGD) with a momentum of 0.9 and weight decay of 5×10^{-4} for 350 epochs.The learning rate is initialized at 0.1 for the first 150 epochs, reduced to 0.01 for the next 100, and further decreased to 0.001 for the final 100 epochs. For ImageNet, we use pretrained models from PyTorch, following the training recipe available on PyTorch's model page.

Evaluation Metrics and Other Settings Calibration performance is primarily evaluated using ECE, with additional metrics including AdaECE, CECE, Negative Log-Likelihood (NLL), and top-1 accuracy. All experiments are conducted on an NVIDIA 4090 GPU, with results averaged over five runs to ensure fairness. For all experiments, we set the number of perturbations to $T = 1000$ and search the perturbation strength ϵ and noise type by minimizing ECE on the validation set. The temperature of TS is likewise selected by minimizing ECE on the validation set.

C Algorithmic Procedure for Consistency Calibration

The procedure below summarizes logit-level Consistency Calibration, which is used as the default CC implementation in the main experiments.

1. Train or load a classifier and compute the logits $\boldsymbol{z}_0 = g(f(x))$ for an input x.
2. Select the perturbation distribution and strength on a validation set by minimizing ECE.
3. For $t = 1, \ldots, T$, sample perturbation noise ϵ_t, form perturbed logits $\widetilde{\boldsymbol{z}}^t = \boldsymbol{z}_0 + \epsilon_t$, and record the winning class $\arg\max_k \widetilde{z}_k^t$.
4. Return the calibrated confidence vector $\hat{\boldsymbol{p}}'$, where each component is the empirical winning frequency of that class across the T perturbations.

D Ablation Studies

Aggregation Methods In our ablation study, we compare two aggregation methods for refining confidence estimates: the mean of softmax probabilities (*Mean*), defined as:

$$\hat{p}_k = \frac{1}{T} \sum_{t=1}^{T} \text{softmax}\left(q\left(\widetilde{h(x)}^t \right) \right), \quad \text{for } k = 1, \ldots, K, \tag{12}$$

and consistency-based aggregation (*Consis.*) as shown in Eq. 7. Both methods leverage predictions over perturbed logits. The mean of softmax probabilities treats the perturbation process like an ensemble method, interpreting uncertainty as a distribution. We show the evaluation results on CIFAR-10 and CIFAR-100 in Table 3. On smaller datasets like CIFAR-10, both methods perform similarly. However, on larger datasets with more classes, such as CIFAR-100 and ImageNet, consistency-based aggregation slightly outperforms softmax averaging. This suggests that consistency-based aggregation captures uncertainty better than the view of ensemble.

Table 3: **Comparison of Aggregation Methods and Noise Types Using ECE↓ Across Various Datasets and Models.** ECE values are reported using 15 bins. The best-performing method for each dataset-model combination is highlighted in bold.

Dataset	Model	Mean U	Mean G	Consis. U	Consis. G
CIFAR-10	ResNet-50	**0.72**	1.34	0.78	1.33
	Wide-ResNet	0.37	0.80	**0.36**	0.83
CIFAR-100	ResNet-50	1.52	2.70	**1.25**	2.49
	Wide-ResNet	1.86	2.08	**1.61**	1.88
ImageNet	ResNet-50	2.37	1.41	2.29	**1.27**
	Wide-ResNet-50	2.17	1.7	2.23	**1.57**

Choice of Noise We investigate the impact of different noise types for input perturbations, comparing uniform noise (*U*) and Gaussian noise (*G*), as shown in Table 3. Uniform noise performs better on datasets with fewer classes, such as CIFAR-10 and CIFAR-100. However, on larger datasets like ImageNet, Gaussian noise yields better results, likely due to variations in the gap between the maximum and second maximum logits across datasets as shown in Figure 6. The choice of noise is treated as a hyperparameter, offering flexibility to adapt to different settings.

Number of Perturbations We also assess the impact of the number of perturbations. As shown in Figure 5, our experiments indicate that CC can achieve good calibration performance with as few as $2^4 = 16$ perturbations. Although increasing the number of perturbations slightly improves results, the diminishing returns suggest that a moderate number of perturbations can provide a practical trade-off between calibration quality and efficiency.

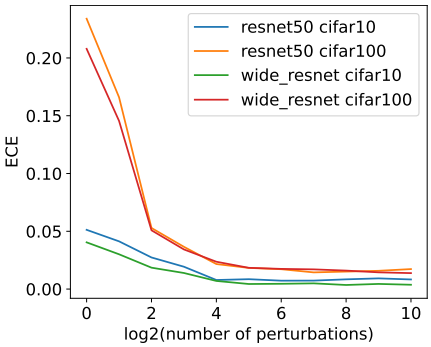


Figure 5: **Calibration Performance with Number of Perturbations**

E Proof of Bounds for Bregman Information

E.1 Preliminaries about the Bregman Information

The Bregman divergence (Gruber & Buettner, 2023) $d_\phi(x, y)$ quantifies the difference between two points, x and y, based on a strictly convex and differentiable function ϕ. It is defined by the formula $d_\phi(x, y) = \phi(y) - \phi(x) - \langle \nabla\phi(x), y - x\rangle$. This value can be understood as the error at point y when using a first-order Taylor approximation of ϕ centered at x. The convexity of ϕ ensures this divergence is always non-negative.

This work focuses on the Bregman Information, $\mathbb{B}_\phi[X]$, which measures the variability of a random variable X using ϕ. It is most intuitively defined as the gap in Jensen's inequality: $\mathbb{B}_\phi[X] = \mathbb{E}[\phi(X)] - \phi(\mathbb{E}[X])$. For differentiable functions, this is equivalent to the expected divergence of X from its own mean: $\mathbb{B}_\phi[X] = \mathbb{E}[d_\phi(\mathbb{E}[X], X)]$. This concept generalizes variance. For instance, when $\phi(x) = x^2$, the Bregman Information $\mathbb{B}_\phi[X]$ is identical to the standard variance of X.

The Bregman Information is relevant to this work because it provides a useful theoretical lens for consistency-based uncertainty measures. Recent studies have identified the Bregman Information of the logits, $\mathbb{B}_{LSE}[\hat{\boldsymbol{z}}]$, as the variance term in a bias-variance decomposition (Gruber & Buettner, 2023) for classifiers, where the randomness stems from training. In our setting, the randomness instead stems from the injected perturbations, so $\mathbb{B}_{\boldsymbol{z}}$ quantifies the dispersion of the perturbed logits for a fixed model rather than model variance. Our theoretical analysis connects this quantity to the Rényi entropy of the consistency-calibrated confidence vector $H_\tau(\boldsymbol{c})$, as reflected by the derived bound $\mathbb{B}_{\boldsymbol{z}} \approx (\tau - 1)H_\tau(\boldsymbol{c})$. This connection supports the use of perturbation-based consistency as an interpretable proxy for logit-level variability under the stated perturbation model.

E.2 Derivation of the Exact Identity

We first evaluate the LSE($\mathbb{E}[\hat{\boldsymbol{z}}]$) term, which can be rewritten as $\mathbb{E}[\hat{\boldsymbol{z}}] = \mathbb{E}[\boldsymbol{z}_0 + \boldsymbol{\epsilon}] = \boldsymbol{z}_0 + \gamma\tau\mathbf{1}$, where the mean of a Gumbel$(0, \tau)$ variable is $\gamma\tau$ and $\gamma \approx 0.577$ is the Euler-Mascheroni constant. Using the LSE's property, we derive $\text{LSE}(\mathbb{E}[\hat{\boldsymbol{z}}]) = \text{LSE}(\boldsymbol{z}_0 + \gamma\tau\mathbf{1}) = \text{LSE}(\boldsymbol{z}_0) + \gamma\tau$. Substituting this into the definition of $\mathbb{B}_{\boldsymbol{z}}$ gives $\mathbb{B}_{\boldsymbol{z}} = \mathbb{E}[\text{LSE}(\hat{\boldsymbol{z}})] - \text{LSE}(\boldsymbol{z}_0) - \gamma\tau$. Then we can define a gap term $G(\hat{\boldsymbol{z}})$ as

$$G(\hat{\boldsymbol{z}}) = \mathbb{E}[\text{LSE}(\hat{\boldsymbol{z}})] - \mathbb{E}[\max(\hat{\boldsymbol{z}})],$$

substituting which into our expression for $\mathbb{B}_{\boldsymbol{z}}$, we get

$$\mathbb{B}_{\boldsymbol{z}} = (\mathbb{E}[\max(\hat{\boldsymbol{z}})] - \text{LSE}(\boldsymbol{z}_0) - \gamma\tau) + G(\hat{\boldsymbol{z}}). \tag{13}$$

Here, we use two standard properties of the Gumbel-Max trick, one is the expected maximum as

$$\mathbb{E}[\max(\hat{\boldsymbol{z}})] = \tau\text{LSE}(\boldsymbol{z}_0/\tau) + \gamma\tau, \tag{14}$$

the other is that it reduces the theoretical consistency to Eq. 10. Substitute Eq. 14 into Eq. 13, we get

$$\mathbb{B}_{\boldsymbol{z}} = (\tau\text{LSE}(\boldsymbol{z}_0/\tau) - \text{LSE}(\boldsymbol{z}_0)) + G(\hat{\boldsymbol{z}}). \tag{15}$$

We now prove the relationship between the bracketed term and $H_\tau(\boldsymbol{c})$. From $\boldsymbol{c} = \text{softmax}(\boldsymbol{z}_0/\tau)$, we have $\ln c_k = z_{0,k}/\tau - \text{LSE}(\boldsymbol{z}_0/\tau)$, which gives:

$$z_{0,k} = \tau(\ln c_k + \text{LSE}(\boldsymbol{z}_0/\tau))$$

Now, substitute this into the definition of $\text{LSE}(\boldsymbol{z}_0)$:

$$\begin{aligned}
\text{LSE}(\boldsymbol{z}_0) &= \ln\left(\sum_k e^{z_{0,k}}\right) = \ln\left(\sum_k e^{\tau(\ln c_k + \text{LSE}(\boldsymbol{z}_0/\tau))}\right) \\
&= \ln\left(\sum_k c_k^\tau \cdot e^{\tau\text{LSE}(\boldsymbol{z}_0/\tau)}\right) \\
&= \ln\left(e^{\tau\text{LSE}(\boldsymbol{z}_0/\tau)} \cdot \sum_k c_k^\tau\right) \\
&= \tau\text{LSE}(\boldsymbol{z}_0/\tau) + \ln\left(\sum_k c_k^\tau\right).
\end{aligned}$$

By the definition (Bromiley et al., 2004), $\ln(\sum_k c_k^\tau) = (1-\tau)H_\tau(\boldsymbol{c})$ where $H_\tau(\cdot)$ is the Rényi entropy, which is a generalized entropy for uncertainty quantification. Rearranging this gives the identity for our bracketed term:

$$\tau\text{LSE}(\boldsymbol{z}_0/\tau) - \text{LSE}(\boldsymbol{z}_0) = (\tau-1)H_\tau(\boldsymbol{c}). \tag{16}$$

Substitute Eq. 16 back into Eq. 15, we prove the following exact identity:

$$\mathbb{B}_{\boldsymbol{z}} = (\tau-1)H_\tau(\boldsymbol{c}) + G(\hat{\boldsymbol{z}}).$$

where $G(\hat{\boldsymbol{z}})$ is a strictly bounded gap term, $0 \le G(\hat{\boldsymbol{z}}) \le \ln K$ (Section E.3), which is small when τ is large relative to the gaps between logits, as for the values of τ in Fig. 3.

E.3 Bounding the Gap Term

The bounds for $\mathbb{B}_{\boldsymbol{z}}$ are determined by the bounds of $G(\hat{\boldsymbol{z}}) = \mathbb{E}[\text{LSE}(\hat{\boldsymbol{z}}) - \max(\hat{\boldsymbol{z}})]$.

Lower Bound ($G(\hat{\boldsymbol{z}}) \ge 0$) For any vector $\boldsymbol{x}$, $\max(\boldsymbol{x})$ is one of the terms in the sum $\sum_k e^{x_k}$. Therefore, $\text{LSE}(\boldsymbol{x}) = \ln(\sum_k e^{x_k}) \ge \ln(e^{\max(\boldsymbol{x})}) = \max(\boldsymbol{x})$. This implies $\text{LSE}(\hat{\boldsymbol{z}}) - \max(\hat{\boldsymbol{z}}) \ge 0$ for any outcome of $\hat{\boldsymbol{z}}$. Taking the expectation, $G(\hat{\boldsymbol{z}}) = \mathbb{E}[\text{LSE}(\hat{\boldsymbol{z}}) - \max(\hat{\boldsymbol{z}})] \ge 0$. Under continuous noise the inequality is strict; the bound is approached in the limit where one perturbed logit dominates all others (e.g., $\tau \to \infty$).

Upper Bound ($G(\hat{\boldsymbol{z}}) \le \ln K$) Let $k^* = \arg\max_i \hat{z}_i$.

$$\begin{aligned}
G(\hat{\boldsymbol{z}}) &= \mathbb{E}\left[\ln\left(\sum_k e^{\hat{z}_k}\right) - \ln\left(e^{\hat{z}_{k^*}}\right)\right] = \mathbb{E}\left[\ln\left(\frac{\sum_k e^{\hat{z}_k}}{e^{\hat{z}_{k^*}}}\right)\right] \\
&= \mathbb{E}\left[\ln\left(\sum_k e^{\hat{z}_k - \hat{z}_{k^*}}\right)\right]
\end{aligned}$$

Since k^* is the index of the maximum, $\hat{z}_k - \hat{z}_{k^*} \le 0$ for all k. Therefore, $e^{\hat{z}_k - \hat{z}_{k^*}} \le e^0 = 1$.

$$G(\hat{\boldsymbol{z}}) \le \mathbb{E}\left[\ln\left(\sum_k 1\right)\right] = \mathbb{E}\left[\ln K\right] = \ln K$$

The bound is approached, but not attained, in the limit where all perturbed logits coincide (equal $\boldsymbol{z}_0$ and $\tau \to 0$).

Table 4: Comparison of calibration error (%) under different noise distributions on ImageNet. CC with Gaussian, Uniform, and Gumbel perturbations achieves consistently lower calibration error than temperature scaling (TS), demonstrating robustness to the choice of noise.

Model	TS	Gaussian	Uniform	Gumbel
ViT-B-16	3.61	1.66	1.81	1.77
ResNet-50	2.09	1.53	1.71	0.87

E.4 Final Statement of Bounds

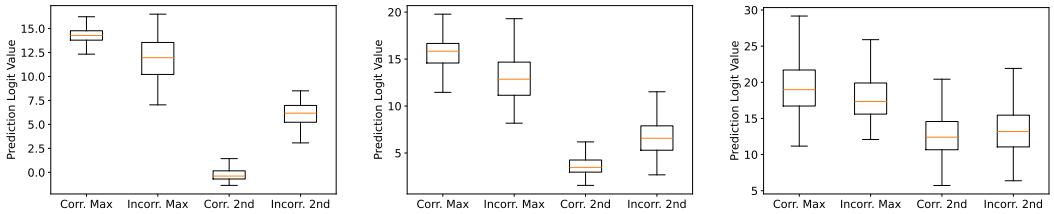
We have rigorously proven the exact identity and the bounds of the gap term:

- **Identity:** $\mathbb{B}_{\boldsymbol{z}} = (\tau - 1)H_\tau(\boldsymbol{c}) + G(\hat{\boldsymbol{z}})$
- **Gap Bounds:** $0 \leq G(\hat{\boldsymbol{z}}) \leq \ln K$

By substituting the lower and upper bounds of $G(\hat{\boldsymbol{z}})$ into the identity, we arrive at the final theorem:

$$(\tau - 1)H_\tau(\boldsymbol{c}) \leq \mathbb{B}_{\boldsymbol{z}} \leq (\tau - 1)H_\tau(\boldsymbol{c}) + \ln K$$

This proves that the Bregman Information of the perturbed logits is bounded by a linear function of the Rényi entropy of the consistency vector, up to an additive gap in $[0, \ln K]$. For $\tau < 1$ the lower bound is negative and hence uninformative, since $\mathbb{B}_{\boldsymbol{z}} \geq 0$ always holds.



(a) CIFAR-10 (b) CIFAR-100 (c) ImageNet-1K

Figure 6: Distribution of the max logit and second-largest logit for correct and incorrect predictions with more than 99% confidence, representing well-calibrated and miscalibrated samples on ResNet-50 across different datasets. The difference between the max logit and second-largest logit is significantly smaller for miscalibrated samples compared to well-calibrated samples.

E.5 Beyond Gumbel Noise

Although Gumbel and Gaussian noises differ in their distributional forms, the effectiveness of CC mainly depends on logit margins rather than the specific noise family. When the noise scale is properly calibrated, different i.i.d. perturbations induce similar rank-flip behavior between competing logits, resulting in comparable calibration effects.

To empirically validate this observation, we evaluate CC with Gumbel noise on ImageNet and compare it with Gaussian and Uniform perturbations. As shown in Table 4, Gumbel noise achieves performance comparable to Gaussian and Uniform noise across both ViT-B-16 and ResNet-50. These results suggest that our framework is largely insensitive to the specific choice of i.i.d. noise distribution, supporting consistency between the theoretical intuition and practical performance.

F Why consistency calibration works?

Perturbing images results in straightforward and intuitive image neighborhoods, but the effectiveness of perturbations at the logit level requires further explanation. To understand why logit perturbations work, we examined the differences between highly confident correct predictions and overconfident incorrect ones. These represent well-calibrated and poorly calibrated samples, respectively. During logit disturbance, the label with second-largest logit most likely to become the prediction label. To

investigate this, we plotted box plots for both the maximum and second-largest logits for correct and incorrect predictions, as shown in Figure 6a.

For CIFAR-10 test samples, we selected predictions with confidence higher than 99%. We refer to the maximum logit of correct predictions as "Corr. Max" and that of incorrect predictions as "Incorr. Max." Similarly, "Corr. 2nd" represents the second-largest logit of correct predictions, while "Incorr. 2nd" refers to the second-largest logit of incorrect predictions. As shown in Figure 6a, the maximum logit for correct predictions is significantly higher than for incorrect predictions. Additionally, the second-largest logit in correct predictions is much lower than that in incorrect predictions. This indicates that the gap between the maximum and second-largest logits is much larger for correct predictions than for incorrect ones. Despite large difference, due to softmax saturation, the model assigns abnormally high confidence (greater than 99%) to both correct and incorrect predictions, leading to overconfident miscalibration.

Interestingly, we can leverage this difference in the logit gaps between correct and incorrect predictions. Perturbations can easily alter the predictions of overconfident, miscalibrated samples, while having minimal effect on well-calibrated, correct predictions. This different response to perturbations explains why consistency calibration is effective at the logit level. We observed similar patterns in experiments with CIFAR-100 and ImageNet, as shown in Figure 6b and Figure 6c.

G PERTURBATION OF DIFFERENT LAYER

This section presents a detailed analysis of the impact of perturbations applied at various levels of a ResNet50 model, trained on CIFAR-10. The experiments were conducted using 32 samples, and the effects on ECE, accuracy, and optimal perturbation values were evaluated.

Table 5: Comparison of perturbations at different layers with number of samples set to 32 using ECE↓ and Accuracy↑, evaluated on ResNet50 with CIFAR-10. ECE values are reported with 15 bins. Optimal Perturbations for logits and features are represented in ϵ value

Perturbation Level	**ECE (%)**	**Accuracy (%)**	**Optimal Perturbation**
Image	1.1	95.25	Train Aug + Jitter (0.1)
Logits	0.73	95.04	8.2
Feature (Last Layer)	2.06	95.06	3.0
Feature (Layer 4)	0.53	95.29	13.28
Feature (Layer 3)	53.12	10.03	20.12
Feature (Layer 2)	56.28	10.02	20.21
Feature (Layer 1)	49.53	10.11	20.75

From Table 5, we observe a clear trend in the performance of perturbations applied at different layers of the model. Perturbation at the logits level achieves a favorable trade-off between calibration and efficiency. Although the perturbation applied to the fourth layer's feature space slightly improves the ECE to 0.53%, the associated computational cost is significantly higher, with the optimal perturbation value of 13.28.

On the other hand, perturbations applied at lower feature levels (Layer 1 to Layer 3) result in severe degradation of both accuracy and calibration. Specifically, the ECE increases drastically to above 50%, and accuracy drops to approximately 10%, with a significant increase in computing time and memory use. This suggests that perturbing the features at these lower layers disrupts the model's ability to recognize patterns and correctly classify the input data. We hypothesize that this is due to the higher sensitivity of lower layers to the raw data structure, where perturbations may significantly distort the features necessary for effective recognition.

H COMPARISON OF POST-HOC CALIBRATION METHODS ON OTHER METRICS

H.1 ADAPTIVE-ECE

As shown in Table 6, the proposed CC obtains low Adaptive-ECE (AdaECE) values across architectures and datasets, indicating improved calibration under this metric. This observation aligns with the findings discussed in the main text. For instance, on CIFAR-10, Wide-ResNet achieves an AdaECE of 0.40 with CC, compared to 3.24 for the uncalibrated baseline (Vanilla), representing a substantial improvement. Similar trends are observed across other datasets and models, suggesting that the effect is not limited to a single setting. The Adaptive-ECE metric is defined as:

$$\text{Adaptive-ECE} = \sum_{i=1}^{B} \frac{|B_i|}{N} \left| A_i - C_i \right|, \tag{17}$$

$$\text{s.t. } \forall i, j; |B_i| = |B_j|, \tag{18}$$

where A_i and C_i denote the accuracy and average confidence within bin B_i, respectively, and N is the total number of samples. Unlike standard ECE, AdaECE employs adaptive binning to ensure each bin contains an equal number of samples, thereby mitigating the bias introduced by uneven confidence distributions.

Table 6: **Comparison of Post-Hoc Calibration Methods Using AdaECE↓.** AdaECE values are reported with 15 bins. The best results for each combination is in bold. Results are averaged over 5 runs.

Dataset	Model	Vanilla	TS	ETS	PTS	CTS	GC	CC (ours)
CIFAR-10	ResNet-50	4.33	2.14	2.14	2.14	1.71	1.24	**0.64**
	ResNet-110	4.40	1.89	1.89	1.90	1.31	**0.94**	0.96
	DenseNet-121	4.49	2.12	2.12	2.12	1.71	1.28	**1.20**
	Wide-ResNet	3.24	1.71	1.71	1.71	1.42	1.17	**0.40**
CIFAR-100	ResNet-50	17.52	5.76	5.72	5.66	5.79	3.43	**1.61**
	Wide-ResNet	15.34	4.48	4.45	4.41	4.69	2.24	**1.73**
ImageNet	ResNet-50	3.73	2.07	2.07	2.06	3.22	2.56	**1.47**
	DenseNet-121	6.59	1.67	1.68	1.69	1.89	2.49	**1.36**
	Wide-ResNet-50	5.32	2.97	2.97	2.95	4.13	2.18	**1.27**
	ViT-B-16	5.59	4.05	4.06	4.08	5.50	1.86	**1.76**
	ViT-B-32	6.40	3.83	3.85	3.91	5.73	**1.33**	1.77

H.2 CLASSWISE-ECE

As shown in Table 7, CC also obtains competitive class-wise calibration results, with lower CECE values than competing methods in most settings. This result suggests that CC can improve calibration not only at the overall level but also across individual classes, as discussed in the main text. For example, on CIFAR-100 with ResNet-50, CC attains a CECE of 0.20, the lowest among all methods evaluated. CECE is a complementary metric to ECE that addresses its limitation of only evaluating the predicted class. It measures calibration quality for each class separately and averages the results across all classes. The formulation is given as:

$$\text{Classwise-ECE} = \frac{1}{\mathcal{K}} \sum_{j=1}^{\mathcal{K}} \sum_{i=1}^{B} \frac{|B_{i,j}|}{N} \left| A_{i,j} - C_{i,j} \right|, \tag{19}$$

where $\mathcal{K}$ is the number of classes, $B_{i,j}$ denotes the i-th confidence bin for class j, and $A_{i,j}$ and $C_{i,j}$ represent the accuracy and average confidence within that bin, respectively.

H.3 NEGATIVE LOG-LIKELIHOOD (NLL)

As shown in Table 8, CC yields higher NLL than TS-based methods in almost all settings, even though it performs well in AdaECE and CECE. The gap is modest on CIFAR (e.g., 108.40 vs.

Table 7: **Comparison of Post-Hoc Calibration Methods Using CECE↓.** CECE values are reported with 15 bins. The best-performing method for each dataset-model combination is in bold. Results are averaged over 5 runs.

Dataset	Model	Vanilla	TS	ETS	PTS	CTS	GC	CC (ours)
CIFAR-10	ResNet-50	0.91	0.45	0.45	0.45	0.41	0.46	**0.39**
	ResNet-110	0.92	0.48	0.48	0.48	0.42	0.52	**0.41**
	DenseNet-121	0.92	0.48	0.48	0.48	**0.41**	0.54	0.43
	Wide-ResNet	0.68	0.37	0.37	0.37	0.37	0.48	**0.32**
CIFAR-100	ResNet-50	0.38	0.21	0.21	0.21	0.22	0.21	**0.20**
	Wide-ResNet	0.34	0.19	0.19	0.19	0.20	0.20	**0.18**
ImageNet	ResNet-50	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
	DenseNet-121	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
	Wide-ResNet-50	0.03	0.03	0.03	0.03	0.03	0.03	**0.02**
	ViT-B-16	0.03	**0.02**	**0.02**	**0.02**	0.03	**0.02**	**0.02**
	ViT-B-32	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**

106.07 for ResNet-50 on CIFAR-100) but large on ImageNet (e.g., 140.57 vs. 94.82 for ResNet-50, an increase of 48%), so it should not be regarded as a small trade-off. We attribute it to how CC constructs the full probability vector: probabilities are empirical frequencies of winning the $\arg\max$ over T perturbations, so classes that rarely win receive (near-)zero probability. NLL heavily penalizes low probability on the true class of misclassified samples, and with 1,000 classes the true class frequently falls in this tail. Top-label metrics such as ECE and AdaECE are insensitive to this tail, which is why CC can improve them while NLL deteriorates. Reducing this effect, e.g., by smoothing the empirical frequencies, is left for future work.

Table 8: **Comparison of Post-Hoc Calibration Methods Using NLL↓.** The best-performing method for each dataset-model combination is in bold. Results are averaged over 5 runs; "–" marks a value that is not available.

Dataset	Model	Vanilla	TS	ETS	PTS	CTS	GC	CC (ours)
CIFAR-10	ResNet-50	41.21	20.39	20.39	20.38	20.15	**19.97**	20.39
	ResNet-110	47.52	21.52	21.52	21.52	20.84	**20.68**	23.33
	DenseNet-121	42.93	21.78	21.78	21.78	21.01	**20.30**	22.19
	Wide-ResNet	26.75	15.33	15.33	15.33	**15.13**	15.32	17.10
CIFAR-100	ResNet-50	153.67	**106.07**	**106.07**	**106.07**	106.25	107.80	108.40
	Wide-ResNet	140.11	**95.71**	**95.71**	**95.71**	96.38	96.92	99.30
ImageNet	ResNet-50	96.12	94.82	94.82	**94.81**	99.58	99.07	140.57
	DenseNet-121	109.52	**103.90**	**103.90**	103.91	106.13	108.14	162.02
	Wide-ResNet-50	88.56	**86.46**	**86.46**	**86.46**	91.68	–	120.59
	ViT-B-16	83.71	**78.63**	**78.63**	**78.63**	85.19	82.14	106.89
	ViT-B-32	107.76	101.67	101.67	**101.66**	107.53	105.45	141.71

H.4 Accuracy

Table 9 shows that the classification accuracy remains largely unchanged across different calibration methods, with all methods performing similarly. This pattern is consistent with the main discussion, indicating that CC improves calibration without compromising predictive accuracy. For instance, on CIFAR-10, Wide-ResNet achieves nearly identical accuracy across all methods, with CC showing marginal improvements in some cases.

Table 9: **Comparison of Post-Hoc Calibration Methods Using Accuracy↑.** Top-1 accuracy values are reported. The best results for each combination is in bold. Results are averaged over 5 runs.

Dataset	Model	Vanilla	TS	ETS	PTS	CTS	GC	CC (ours)
CIFAR-10	ResNet-50	95.05	95.05	95.05	95.05	94.98	95.05	**95.06**
	ResNet-110	95.11	95.11	95.11	95.11	**95.18**	95.11	95.16
	DenseNet-121	95.02	95.02	95.02	95.02	95.01	95.02	**95.04**
	Wide-ResNet	**96.13**	**96.13**	**96.13**	**96.13**	96.06	**96.13**	**96.13**
CIFAR-100	ResNet-50	76.70	76.70	76.70	76.70	**76.72**	76.70	76.71
	Wide-ResNet	79.29	79.29	79.29	79.29	79.17	79.29	**79.31**
ImageNet	ResNet-50	**76.08**	**76.08**	**76.08**	**76.08**	74.62	**76.08**	**76.08**
	DenseNet-121	74.16	74.16	74.16	74.16	73.08	74.16	**74.37**
	Wide-ResNet-50	78.40	78.40	78.40	78.40	77.07	78.40	**78.48**
	ViT-B-16	**81.09**	**81.09**	**81.09**	**81.09**	80.01	**81.09**	81.06
	ViT-B-32	**75.94**	**75.94**	**75.94**	**75.94**	74.90	**75.94**	75.90

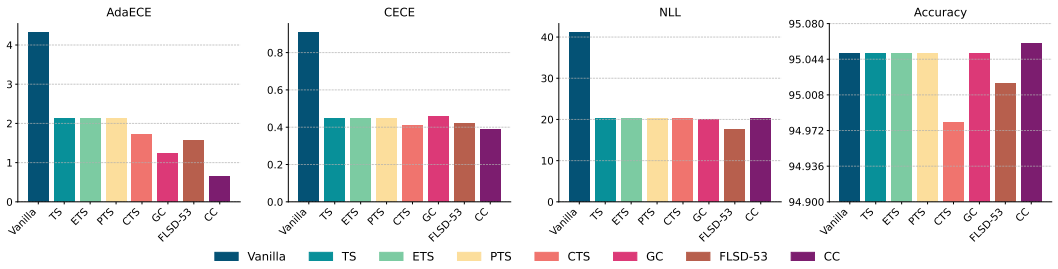


Figure 7: **Calibration performance of ResNet-50 on Cifar-10 using AdaECE↓, CECE↓, NLL↓, and Accuracy↑.** ECE, AdaECE, and CECE are reported with 15 bins. Colors in the legend represent different methods. Results are averaged over 5 runs.

I Comparison of various training-time calibration methods on other metrics

I.1 Adaptive-ECE

As shown in Table 10, the proposed Consistency Calibration (CC) method reduces AdaECE across datasets and architectures. On CIFAR-10 and CIFAR-100, CC achieves lower AdaECE scores for ResNet-50, ResNet-110, DenseNet-121, and Wide-ResNet compared with several training-time calibration methods such as Brier Loss and MMCE. For instance, on CIFAR-100 with ResNet-110, CC reduces the AdaECE from 19.05 (baseline) to 5.28, indicating improved calibration while maintaining model performance.

Table 10: **Comparison of Train-time Calibration Methods Using AdaECE↓.** AdaECE values are reported with 15 bins. The best results for each combination is in bold. Results are averaged over 5 runs.

Dataset	Model	Cross-Entropy		Brier Loss		MMCE		LS-0.05		FLSD-53		FL-3	
		base	ours	base	ours	base	ours	base	ours	base	ours	base	ours
CIFAR-10	ResNet-50	4.33	**0.64**	1.75	**0.99**	4.55	**1.06**	3.88	**1.74**	1.56	**0.36**	1.95	**0.71**
	ResNet-110	4.40	**0.96**	2.60	**0.30**	5.07	**1.80**	4.48	**2.43**	2.08	**0.73**	1.64	**0.38**
	DenseNet-121	4.49	**1.20**	2.02	**0.64**	5.10	**1.76**	4.40	**1.94**	1.38	**0.53**	1.23	**0.69**
	Wide-ResNet	3.24	**0.40**	1.70	**0.57**	3.29	**0.63**	4.27	**1.54**	1.52	**0.42**	1.84	**0.42**
CIFAR-100	ResNet-50	17.52	**1.61**	6.55	**1.90**	15.32	**1.88**	7.66	**6.17**	4.39	**1.48**	5.09	**1.70**
	ResNet-110	19.05	**5.28**	7.72	**3.54**	19.14	**5.14**	11.14	**8.00**	8.56	**3.50**	8.64	**3.98**
	DenseNet-121	20.99	**5.85**	5.04	**2.02**	19.10	**3.90**	12.83	**7.06**	3.54	**1.52**	4.14	**2.03**
	Wide-ResNet	15.34	**1.73**	4.28	**1.92**	13.16	**2.06**	5.14	**4.75**	2.77	**1.79**	2.07	**1.58**

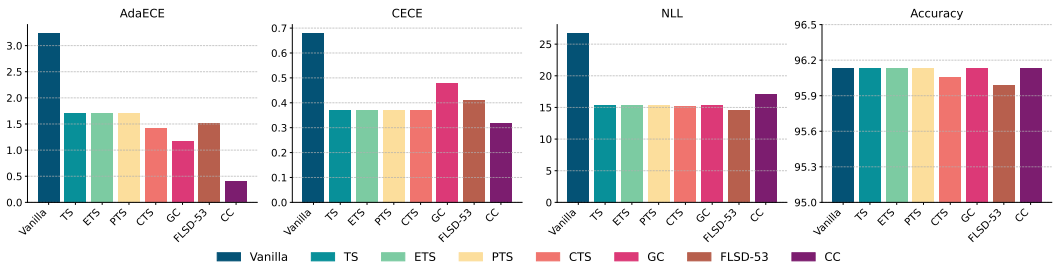


Figure 8: **Calibration performance of Wide-ResNet on CIFAR-10 using AdaECE↓, CECE↓, NLL↓, and Accuracy↑.** ECE, AdaECE, and CECE are reported with 15 bins. Colors in the legend represent different methods. Results are averaged over 5 runs.

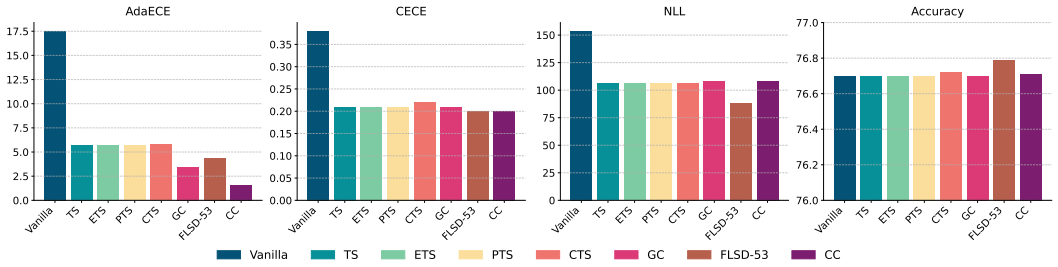


Figure 9: **Calibration performance of ResNet-50 on CIFAR-100 using AdaECE↓, CECE↓, NLL↓, and Accuracy↑.** ECE, AdaECE, and CECE are reported with 15 bins. Colors in the legend represent different methods. Results are averaged over 5 runs.

I.2 Class-wise ECE

As shown in Table 11, the CECE results provide additional evidence for the effectiveness of CC across different architectures and datasets. On CIFAR-10, CC improves CECE for all models compared with the baseline methods. For example, for ResNet-50, CECE decreases from 0.91 to 0.39. Similar trends are observed on CIFAR-100, where Wide-ResNet shows a reduction from 0.34 (baseline) to 0.18 when using CC, indicating improved class-wise calibration performance.

Table 11: **Comparison of Train-time Calibration Methods Using CECE↓.** CECE values are reported with 15 bins. The best results for each combination is in bold. Results are averaged over 5 runs.

Dataset	Model	Cross-Entropy		Brier Loss		MMCE		LS-0.05		FLSD-53		FL-3	
		base	ours	base	ours	base	ours	base	ours	base	ours	base	ours
CIFAR-10	ResNet-50	0.91	**0.39**	0.46	**0.35**	0.94	**0.47**	0.71	**0.53**	0.42	**0.35**	0.43	**0.39**
	ResNet-110	0.92	**0.41**	0.59	**0.41**	1.04	**0.50**	**0.66**	0.67	0.48	**0.39**	0.43	**0.37**
	DenseNet-121	0.92	**0.43**	0.46	**0.37**	1.04	**0.59**	0.60	**0.48**	0.41	**0.35**	0.42	**0.35**
	Wide-ResNet	0.68	**0.32**	0.44	**0.32**	0.70	**0.38**	0.79	**0.41**	0.41	**0.28**	0.44	**0.30**
CIFAR-100	ResNet-50	0.38	**0.20**	0.22	**0.19**	0.34	**0.18**	0.23	**0.22**	0.20	**0.19**	0.20	**0.19**
	ResNet-110	0.41	**0.21**	0.24	**0.19**	0.42	**0.20**	0.26	**0.22**	0.24	**0.19**	0.24	**0.20**
	DenseNet-121	0.45	**0.23**	0.20	0.20	0.42	**0.23**	0.29	**0.22**	0.19	0.19	0.20	**0.19**
	Wide-ResNet	0.34	**0.18**	0.19	**0.18**	0.30	**0.17**	0.21	**0.19**	0.18	**0.17**	0.18	**0.17**

I.3 Negative Log-Likelihood (NLL)

Table 12 presents the NLL comparison. It is interesting as mentioned in the main section, the CC method sometimes produces higher NLL values.

Table 12: **Comparison of Train-time Calibration Methods Using NLL↓.** The best-performing method for each dataset-model combination is in bold. Results are averaged over 5 runs.

Dataset	Model	Cross-Entropy		Brier Loss		MMCE		LS-0.05		FLSD-53		FL-3	
		Base	Ours	Base	Ours	Base	Ours	Base	Ours	Base	Ours	Base	Ours
CIFAR-10	ResNet-50	41.2	**20.4**	**18.7**	22.3	44.8	**20.9**	**27.7**	29.3	**17.6**	22.7	**18.4**	24.2
	ResNet-110	47.5	**25.5**	**20.4**	22.5	55.7	**25.5**	29.9	**29.4**	**18.5**	21.9	**17.8**	23.1
	DenseNet-121	42.9	**24.0**	**19.1**	21.2	52.1	**31.2**	28.7	**28.5**	**18.4**	27.2	**18.0**	28.3
	Wide-ResNet	26.8	**17.1**	**15.9**	16.2	28.5	**18.2**	**21.7**	24.5	**14.6**	17.6	**15.2**	19.9
CIFAR-100	ResNet-50	153.7	**113.0**	**99.6**	133.5	125.3	**116.7**	**121.0**	133.9	**88.0**	128.8	**87.5**	128.1
	ResNet-110	179.2	**122.3**	**110.7**	146.9	180.6	**125.3**	**133.1**	141.4	**89.9**	126.9	**90.9**	132.0
	DenseNet-121	205.6	**163.1**	**98.3**	139.9	166.6	**146.8**	**142.0**	185.8	**85.5**	129.0	**87.1**	130.8
	Wide-ResNet	140.1	**102.5**	**84.6**	98.7	119.6	**109.3**	**108.1**	136.6	**76.9**	108.7	**74.7**	106.8

I.4 ACCURACY

Table 13 presents a comparison of classification accuracies. While achieving superior calibration performance by CC, the accuracy remains unaffected across all metrics.

Table 13: **Comparison of Train-time Calibration Methods Using Accuracy↑.** Top-1 Accuracy values are reported. The best results for each combination is in bold. Results are averaged over 5 runs.

Dataset	Model	Cross-Entropy		Brier Loss		MMCE		LS-0.05		FLSD-53		FL-3	
		base	ours	base	ours	base	ours	base	ours	base	ours	base	ours
CIFAR-10	ResNet-50	95.05	**95.06**	94.99	**95.01**	**95.01**	94.99	**94.71**	94.68	**95.02**	94.95	94.75	94.75
	ResNet-110	95.11	**95.16**	**94.52**	94.48	94.60	**94.63**	94.48	**94.49**	94.57	**94.63**	94.92	**94.94**
	DenseNet-121	**95.02**	95.01	**94.90**	94.86	94.59	**94.60**	94.91	94.91	**94.58**	94.51	94.66	94.66
	Wide-ResNet	**96.13**	96.12	**95.92**	95.90	**96.09**	96.05	95.80	**95.83**	95.99	**96.01**	95.87	95.87
CIFAR-100	ResNet-50	76.70	**76.71**	**76.60**	76.58	76.80	76.80	76.56	**76.65**	**76.79**	76.73	77.24	**77.34**
	ResNet-110	**77.27**	77.17	**74.91**	74.79	76.93	**76.96**	76.57	**76.64**	77.48	**77.49**	**77.08**	77.04
	DenseNet-121	75.47	**75.49**	76.27	**76.30**	76.03	76.03	75.94	**75.96**	77.34	77.34	76.76	**76.85**
	Wide-ResNet	**79.29**	79.25	**79.43**	79.29	**79.27**	79.23	78.83	**78.88**	79.91	**79.92**	80.30	**80.34**