Efficient Post-Hoc Uncertainty Calibration via Variance-Based Smoothing

Abstract

Since state-of-the-art uncertainty estimation methods are often computationally demanding, we investigate whether incorporating prior information can improve uncertainty estimates in conventional deep neural networks. Our focus is on machine learning tasks where meaningful predictions can be made from sub-parts of the input. For example, in speaker classification, the speech waveform can be divided into sequential patches, each containing information about the same speaker. We observe that the variance between sub-predictions serves as a reliable proxy for uncertainty in such settings. Our proposed variance-based scaling framework produces competitive uncertainty estimates in classification while being less computationally demanding and allowing for integration as a post-hoc calibration tool. This approach also leads to a simple extension of deep ensembles, improving the expressiveness of their predicted distributions. The code to replicate this work is available via GitHub.

1 INTRODUCTION

Deep neural networks (NNs) have achieved remarkable performance across a wide range of machine learning tasks and are increasingly used in critical applications [Krizhevsky et al., 2012, Ioffe, 2015, Miotto et al., 2018]. However, while they excel in predictive accuracy, they often struggle to quantify uncertainty reliably, leading to overconfident predictions [Lakshminarayanan et al., 2017, Liu et al., 2023, Guo et al., 2017]. This is a serious problem in high-stakes domains such as healthcare and autonomous systems, where well-calibrated confidence estimates are crucial. A perfectly calibrated model, for example, would predict “cat” with 70% confidence and be correct 70% of the time.

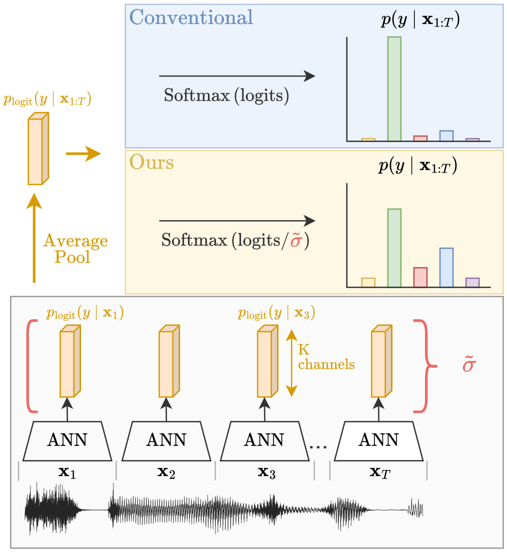
Conventional
$p(y \mid \mathbf{x}_{1:T})$
Softmax (logits)
$p_{\text{logit}}(y \mid \mathbf{x}_{1:T})$
Ours
$p(y \mid \mathbf{x}_{1:T})$
Softmax (logits/$\tilde{\sigma}$)
Average Pool
$p_{\text{logit}}(y \mid \mathbf{x}_1)$
$p_{\text{logit}}(y \mid \mathbf{x}_3)$
K channels
$\tilde{\sigma}$
ANN
ANN
ANN
ANN
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{x}_T$

Figure 1: Overview of our method. We use a pre-trained classifier and variance between logits of sub-patches as an uncertainty proxy to recalibrate uncertainty estimates.

State-of-the-art methods for improving uncertainty estimation, such as deep ensembles, Bayesian neural networks, and MC-dropout [Lakshminarayanan et al., 2017, Blundell et al., 2015, Gal and Ghahramani, 2016], have shown strong results. However, these methods come with substantial computational costs. Ensembles require training multiple models, increasing both memory usage and inference time. Bayesian neural networks, which learn distributions over weights, suffer from slow training and expensive inference. MC-dropout, while lighter than ensembles, still requires multiple stochastic forward passes during inference and keeping dropout active at test time, making it less suitable for real-time applications. These constraints limit the practicality of existing uncertainty quantification (UQ) methods in resource-constrained environments, such as autonomous drones or embedded systems.

Interestingly, many real-world tasks involve input data with inherent redundancy, where the problem can be solved using substructures or subsets of the input. For instance, in speech classification, short audio patches often contain enough information to determine the speaker. Similarly, in radio signal identification, brief signal windows can be sufficient to identify the transmitting device. Despite this, existing UQ methods do not explicitly exploit such redundancy, leaving an opportunity for more efficient approaches.

In this work, we propose a novel variance-based softmax scaling approach to recalibrate a model's uncertainty estimates by exploiting tasks that follow what we refer to as the "informative sub-patches" assumption. Specifically, we introduce a method that uses the variance between predictions on sub-patches of the input as a proxy for uncertainty, achieving real-time inference speeds while improving the model's reliability. We refer to this approach as Variance-based Smoothing (Fig. 1).

Our contributions are threefold:

1. We propose a simple yet effective recalibration method that serves as a post-hoc uncertainty estimation technique for pre-trained models while preserving their original accuracy.
2. We demonstrate competitive uncertainty estimates on both clean and noisy data for diverse tasks, including speech, vision, and radio signal classification while being significantly more efficient.
3. Our proposal also leads to a simple extension of ensembles, improving upon the expressiveness of their predicted distributions in tasks with many classes while eliminating the need for the informative sub-patches assumption.

2 EXPLOITING SUBSTRUCTURE FOR UNCERTAINTY ESTIMATION

This section is structured as follows: we begin with a naive approach to exploiting this *informative sub-patches assumption* and discuss its limitations, followed by a refined method that preserves classification performance. Finally, we consider a generalization of this approach as an extension of ensembles, allowing us to omit this data requirement.

2.1 ENSEMBLE MODELS AS A STARTING POINT: A NAIVE APPROACH

Ensemble-based methods train multiple models with different weight initializations. When the same input is passed through these models, their predictions tend to agree on in-distribution data similar to what they were trained on. However, for out-of-distribution data, predictions may vary due to differences in the learned model weights. The final classification output is computed as the mean of the softmax predictions:

$$p(y \mid \mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} p_{\theta_m}(y \mid \mathbf{x}), \tag{1}$$

where $p(y \mid \mathbf{x})$ is a probability vector over K classes, M is the number of models, and θ_m denotes the parameters of the m-th model [Lakshminarayanan et al., 2017]. The predictive distribution also serves as the uncertainty estimate.

Based on these ensembles, a naive single-model alternative that exploits the *informative sub-patches assumption* is to split an input $\mathbf{x}$ into T (non)overlapping sub-patches $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_T$ and make separate predictions on each sub-patch individually. The final prediction is obtained by averaging the softmax distributions:

$$p(y \mid \mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} p(y \mid \mathbf{x}_t). \tag{2}$$

Unlike conventional NNs, where logits are pooled before applying softmax, this approach averages already-softmaxed outputs. While this can temper overconfident predictions, it also harms predictive accuracy, as each prediction is based on a sub-patch with a limited context window. Additionally, if a given sub-patch $\mathbf{x}_t$ contains little information (e.g., a 500ms audio segment where the speaker is silent), the model may output unconfident logit scores where all class values are negative. After applying softmax, a peak is still "forced," causing the prediction from that sub-patch to contribute equally to those from more informative patches. We found that this method tends to produce overly unconfident predictions for both in-distribution and out-of-distribution data, resulting in less informative output distributions.

2.2 VARIANCE-BASED SMOOTHING

To address these issues, we preserve the conventional pre-softmax classification output, denoted as $p_{\text{logit}}(y \mid \mathbf{x})$, and adjust softmax confidence using temperature scaling. The scaling factor $\tilde{\sigma} \geq 1$, derived from the variance of sub-predictions, adjusts the softmax function to better account for uncertainty:

$$p(y \mid \mathbf{x}) = \text{Softmax}\left(\frac{p_{\text{logit}}(y \mid \mathbf{x})}{\tilde{\sigma}}\right). \tag{3}$$

Since temperature scaling does not affect the predicted class ranking, classification accuracy remains unchanged.

To obtain $\tilde{\sigma}$, the standard deviation σ_k is computed across sub-predictions for class k over different sub-patch-indices t:

$$\sigma_k := \sqrt{\frac{1}{T-1} \sum_{t=1}^{T} \left(\left[p_{\text{logit}}\ (y \mid \mathbf{x}_t) \right]_k - \mu_k \right)^2}, \tag{4}$$

where $\mu_k = \frac{1}{T}\sum_{t=1}^{T} \left[p_{\text{logit}}\ (y \mid \mathbf{x}_t)\right]_k$. We aggregate the per-class standard deviations $\sigma_1, \ldots, \sigma_K$ into a single scalar $\bar{\sigma}$ and compute $\tilde{\sigma}$ as follows:

$$\bar{\sigma} = \frac{1}{K}\sum_{k=1}^{K} \sigma_k, \qquad \tilde{\sigma} = \max(\alpha(\bar{\sigma} + \beta), 1). \tag{5}$$

The max operation ensures that $\tilde{\sigma}$ never falls below 1, preventing unintended confidence amplification. The hyperparameters α and β control the sensitivity of scaling. We found empirically that reasonable values for β are between the negative 50th and 95th percentiles of $\bar{\sigma}$ computed over a validation set, ensuring that smoothing activates only beyond a certain threshold. The parameter α determines the strength of the adjustment, with $\alpha = 1$ providing balanced uncertainty estimates. Higher values can be used to encourage more conservative or pessimistic predictions, with values between 1 and 5 being reasonable choices.

Overall, this Variance-based Smoothing approach requires only a single model and a single forward pass over the entire input, significantly reducing computational costs compared to ensembles or MC-dropout approaches. The additional variance computation is minimal, making it practical for real-time applications and useful as a post-hoc calibration tool for existing models. The variance computation, however, requires the architecture to preserve spatial information in the final logits, which typically many convolution-based architectures do.

2.3 VARIANCE-BASED SMOOTHING AS AN EXTENSION OF ENSEMBLES

The proposed Variance-based Smoothing procedure in Sec. 2.2 requires a variance source, which it obtains from sub-predictions across *informative* sub-patches. However, this data-requirement and potential restrictions to the architecture can be omitted when an alternative source of variance is available. One such source is the variance between ensemble predictions, specifically, their logits.

In this case, the temperature value $\tilde{\sigma}$ is obtained by averaging and rescaling σ_k, now defined as the standard deviation of the predicted logits across the models for class k:

$$\sigma_k := \sqrt{\frac{1}{M-1}\sum_{m=1}^{M} \left(\left[p_{\text{logit}\ \theta_m}\ (y \mid \mathbf{x})\right]_k - \mu_k\right)^2}, \tag{6}$$

where $\mu_k = \frac{1}{M}\sum_{m=1}^{M} \left[p_{\text{logit}\ \theta_m}\ (y \mid \mathbf{x})\right]_k$. The final distribution is then computed as:

$$p(y \mid \mathbf{x}) = \text{Softmax}\left(\frac{\frac{1}{M}\sum_{m}^{M} p_{\text{logit}\ \theta_m}\ (y \mid \mathbf{x})}{\tilde{\sigma}}\right). \tag{7}$$

This formulation extends ensemble methods while better preserving relative class importance within a single model. Conventional ensembles apply softmax independently to each model's predictions, often producing distributions with sharp peaks that may obscure relative confidence levels [Hinton et al., 2015, Park et al., 2019, Lakshminarayanan et al., 2017]. Moreover, when the number of classes is large, a small ensemble may be unable to represent certain distributions, such as the uniform distribution, because there are simply not enough distinct prediction peaks across models to cover all classes evenly. This limitation is empirically demonstrated in Sec. 3.4, where we show how variance-based smoothing enables ensembles to express higher-entropy distributions and approximate uniform outputs more effectively.

3 EXPERIMENTS

Our experiments are designed to answer the following research questions: (1) How does Variance-based Smoothing affect calibration and confidence under dataset shifts for datasets that satisfy the informative sub-patches assumption versus a dataset with a weaker form? (2) How do its computational costs compare against related methods? (3) How are uncertainty distributions represented when applied to ensembles with a high number of class labels?

3.1 SETUP: DATASETS AND PROTOCOL

We evaluate our method on three publicly available datasets. Two satisfy the *informative sub-patches assumption*, such that their inputs can be split into sub-patches that preserve sufficient task-relevant information. The first is the Radio Signals dataset [Nhem et al., 2025] with 20 class labels, consisting of the ID of the device that emitted the signal. The second is the 100-hour LibriSpeech subset by Löwe et al. [2019], consisting of spoken utterances from 251 speakers, based on Panayotov et al. [2015]. Lastly, we include the CIFAR-10 dataset [Krizhevsky and Hinton, 2009], which contains a much weaker form of this sub-patch assumption; e.g., based on the upper-left patch in the image, it is much more difficult to make an informed prediction due to insufficient context. Model architectures and training protocols follow those in the respective dataset papers (Appendix A).

Baselines: We compare our lightweight post-hoc method against the following methods: a conventional classifier, MC-dropout [Gal and Ghahramani, 2016] (which applies dropout ($p = 0.5$) after every ReLU during both training and inference), Temperature Scaling [Guo et al., 2017] (a post-hoc method that learns a fixed temperature scalar from the validation set for improved calibration), and Deep Ensembles [Lakshminarayanan et al., 2017] (without adversarial training). Both post-hoc methods and the ensemble reuse the weights from the conventional model. Details in Appendix A.5.

Input Splitting and Sub-Patch Processing: The input $\mathbf{x}$ is implicitly split into sub-patches $\mathbf{x}_1, \ldots, \mathbf{x}_T$ based on the ConvNet's architecture. Convolutional layers with specific kernel sizes and strides control the context size and overlap of these sub-patches. We capture logits $\mathbf{z} \in \mathbb{R}^{T \times K}$ before the final pooling layer (architecture details are discussed in Appendix A). Each $\mathbf{z}_t \in \mathbb{R}^K$ represents a prediction for sub-patch $\mathbf{x}_t$, previously denoted as $p_{\text{logit}}(y \mid \mathbf{x}_t)$ in Eq. 4.

To ensure sufficient context, we "merge" neighboring logits $\mathbf{z}_t \ldots \mathbf{z}_{t+j}$ by applying average pooling across $\mathbf{z}$ with a specific kernel size where necessary. This process ensures that each $\mathbf{z}_t$ captures a broader spatial or temporal region. The variance scalar $\tilde{\sigma}$ is then computed based on these pooled logits $\mathbf{z}_1 \ldots \mathbf{z}_{T'}$ rather than the original logits. For Radio Signals, the input $\mathbf{x} \in \mathbb{R}^{1580 \times 2}$ is reduced to $\mathbf{z} \in \mathbb{R}^{79 \times 20}$. A window-pooling operation (kernel size 10, stride 1) is applied, resulting in 70 pooled logits per sample. For LibriSpeech, the input $\mathbf{x} \in \mathbb{R}^{20480 \times 1}$ is reduced to $\mathbf{z} \in \mathbb{R}^{128 \times 251}$, with average pooling (kernel size 40) producing a $\mathbb{R}^{89 \times 251}$ pooled tensor. For CIFAR-10, the input $\mathbf{x} \in \mathbb{R}^{32 \times 32 \times 3}$ is reduced to $\mathbf{z} \in \mathbb{R}^{8 \times 8 \times 10}$, where standard deviations are computed across the 64 logits without additional pooling.

The values of α and β are tuned per dataset, balancing entropy levels across different noise levels while maintaining reliable calibration (Sec. 3.2). For Radio Signals and CIFAR-10, we set $\alpha = 1$ and $\beta = \text{Avg}(\bar{\sigma}) + 0.5$ where $\text{Avg}(\bar{\sigma})$ is the mean of all $\bar{\sigma}$ values computed on the validation set. For LibriSpeech, we use $\alpha = 5$ and set β to the negative 95th percentile of $\bar{\sigma}$ values. In principle, one could also learn α and β by minimizing the cross-entropy loss on the validation set (with a frozen backbone, as done in the Temperature Scaling baseline).

3.2 CALIBRATION AND DATASET SHIFT

3.2.1 Reliability Diagrams on Clean Samples

To evaluate model calibration, we follow the approach of Guo et al. [2017], computing reliability diagrams based on a clean test set. Each prediction is assigned to one of ten confidence bins based on the probability score of their most likely class. The accuracy is then computed for all predictions within a bin. The bins are split into intervals of 10%: $[0\%, 10\%), [10\%, 20\%) \ldots [90\%, 100\%)$, apart from LibriSpeech where bin sizes are intervals of 20% due to an otherwise insufficient amount of data points per bin. A perfectly calibrated model would produce the identity line.

Fig. 2 displays the results for each dataset. For the Radio and CIFAR-10 datasets, all four models follow the identity line, with the conventional NN consistently being overconfident, particularly in the radio's $[80\%, 90\%)$ confidence bin, where the actual accuracy is approximately 65%. MC-dropout generally produces more calibrated results than conventional but is consistently underconfident (as most of its box plots are above the perfect calibration line). Both Temperature Scaling and Variance-based Smoothing seem to significantly improve calibration on Radio, bringing the conventional NN's predictions closer to the identity line. Interestingly for CIFAR-10, Temperature Scaling seems to sharpen distributions rather than smoothening them, resulting in increased confidence and worse calibration scores. Overall, Variance-based Smoothing offers very competitive calibration scores on both RADIO and CIFAR-10.

For LibriSpeech, calibration follows a similar trend: the conventional model tends to be overconfident, MC-dropout is underconfident, and Variance-based Smoothing falls in between, with box plots both above and below the perfect calibration line. Temperature Scaling performs overall closest to the perfect calibration line. However, all four methods produce wider accuracy variations across runs further away from the perfect calibration line, stressing the difficulty of this 251-class dataset.

Variance-based Smoothing produces some improvement in calibrating the conventional NN, with the 50th percentile of its box plots generally closer to the calibration line, though the effect is less pronounced than in the other datasets. Nonetheless, in the LibriSpeech setting, its value is somewhat limited as the method produces many unconfident but perfectly correct predictions, as indicated by the high empirical accuracy in the $[0, 10\%)$ bin, resulting in overly pessimistic predictions within this specific confidence range.

More detailed figures including the graphs on the number of samples produced in each bin and individual per-method reliability diagrams are available in Appendix B.

3.2.2 Confidence under Dataset Shift

Following a modified version of Snoek et al. [2019], we study how confidence evolves under increasing noise perturbations. We consider Gaussian and Speckle noise for Radio and LibriSpeech, while CIFAR-10 is tested under Gaussian, distortion, and affine transformation noise (details in Appendix A.4). This section first evaluates variance as a proxy for uncertainty in the context of Variance-based Smoothing, followed by an evaluation of confidence across all methods.

Variance as a Proxy for Predictive Uncertainty: To empirically validate whether the standard deviation across subpatch predictions ($\bar{\sigma}$) serves as a reliable metric for uncertainty, we incrementally introduce noise and observe how $\bar{\sigma}$ changes. A strong correlation is important, as it is the key informative factor in Variance-based Smoothing.

The results in Fig. 3 show that on the Radio dataset, as noise increases (and accuracy decreases), $\bar{\sigma}$ increases monotonically at two different rates, in line with the accuracy drop

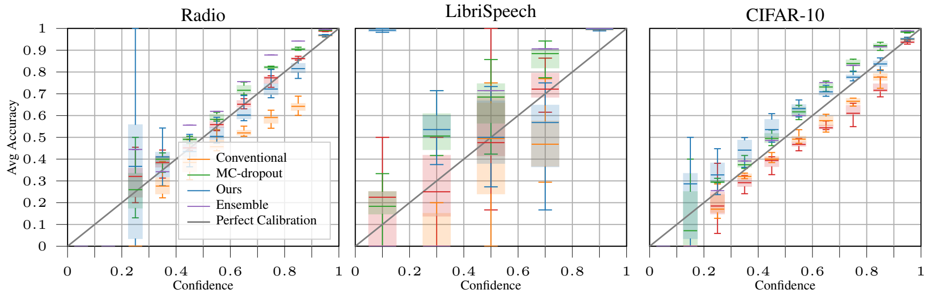

Figure 2: Radio, Librispeech, and CIFAR-10: Reliability diagrams on a clean test set. Variance-based Smoothing improves the calibration of the conventional NN on the RADIO and CIFAR-10 datasets. Individual plots are available in Appendix B.

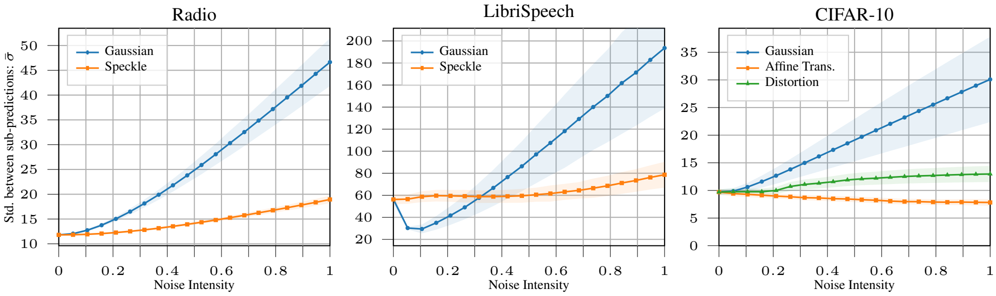

Figure 3: Radio, LibriSpeech and, CIFAR-10: mean standard deviation ($\bar{\sigma}$) across increasingly higher noise intensities.

for each noise type (Fig. 4 and 5). This suggests that variance between sub-patch predictions is a reliable uncertainty proxy in this setting.

For LibriSpeech, while speckle noise produces $\bar{\sigma}$ values that increase proportional to the model's accuracy drop (Fig. 4), Gaussian noise shows an initial dip before recovering, which is suboptimal. The cause of this dip remains unclear. We initially hypothesized that it was due to the number of classes (251) and the nonlinear punishment of softmax in cross-entropy loss on logits, which can cause large variance values for certain logits. For instance, consider two sets of predicted logits: $\begin{pmatrix} 17 & 0.05 & 0.01 & -0.05 \end{pmatrix}$ and $\begin{pmatrix} 17 & -20 & -20 & -20 \end{pmatrix}$, where the first index corresponds to the correct label. Despite their large numerical differences, these logits produce similar cross-entropy losses because the softmax function pushes the negative and near 0 values toward zero. As a result, the variance values σ_k for $2 \leq k \leq 4$ become high. However, training on a 10-class subset resulted in similar behavior, suggesting that class count is not the root cause. We also tested whether sub-patch length played a role, but experiments with varying context windows and filtering for long audio samples all preserved this initial dip.

For CIFAR-10, the method generalizes surprisingly well despite the weak informative sub-patches assumption (evidenced by an average sub-patch accuracy of 0.27 ± 0.013). The variances growths observed for Gaussian and distortion noise are proportional to their respective accuracy drops (Fig. 5), indicating potential applications beyond the assumed dataset assumptions. However, the affine transformation noise results in a negative correlation, which shows an important limitation: sub-patches must be sufficiently different from each other for variance to serve as a reliable proxy. Otherwise, variance remains artificially low despite high uncertainty.

Predictive Entropy: We now analyze confidence across methods using predictive entropy under dataset shift (Radio and LibriSpeech: Fig. 4, and CIFAR-10: Fig. 5).

Across all datasets, both the conventional NN and Temperature-Scaling show some uncertainty at low noise levels, as seen in their increasing entropy. However, at higher noise levels, both methods becomes overly confident despite being incorrect, showing the need for better post-hoc recalibration.

On the Radio dataset, Variance-based Smoothing produces entropy trends that align well with accuracy drops, performing on par with MC-dropout across both noise types.

For LibriSpeech, despite the initial drop in standard deviation (Fig. 3), Variance-based Smoothing still improves uncertainty estimates over the conventional NN, particularly at higher Gaussian noise levels. Importantly, due to the max operation in Eq. 5, our approach suffers less in regions where variance is less reliable. For Speckle noise, Variance-based Smoothing provides consistently increasing entropy levels, whereas the conventional model flattens around a noise level of 0.6. Nonetheless, MC-dropout remains the

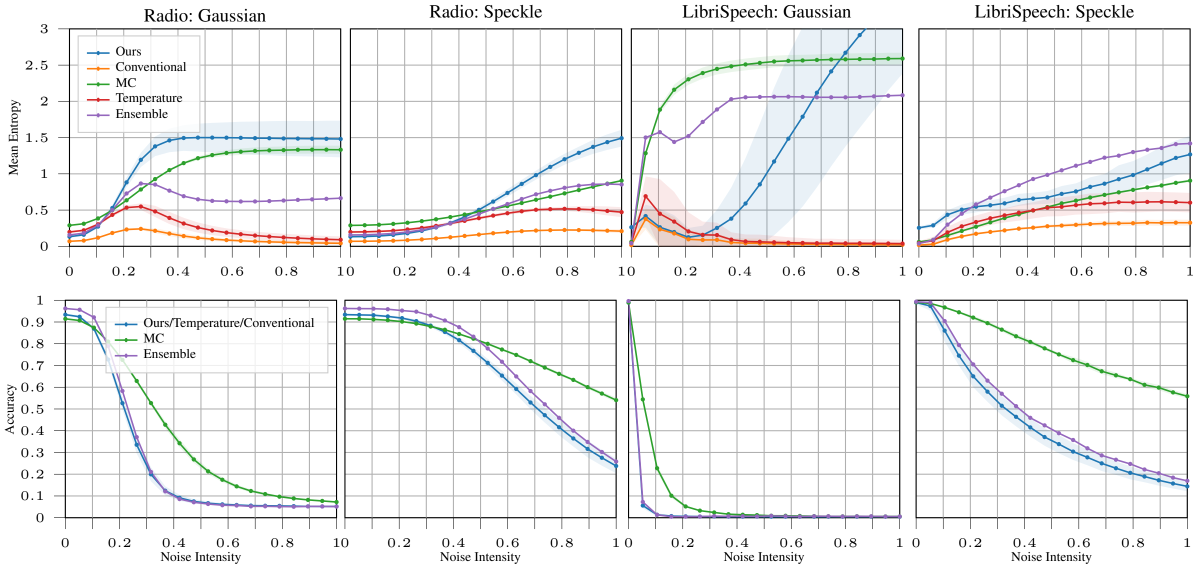

Figure 4: **Radio and LibriSpeech**: Mean entropy and their respective accuracies across different noise intensities. Our approach and Temperature Scaling have the same accuracy as conventional.

better choice for LibriSpeech.

On CIFAR-10, entropy results are similar to those in the Radio dataset. Variance-based Smoothing significantly improves uncertainty estimates for Gaussian and distortion noise, even outperforming MC-dropout. Interestingly, MC-dropout shows an entropy drop at higher noise levels, resulting in varying entropy scores. This suggests that some runs are more confident under noise than on clean data, an undesirable outcome. Continuing with Variance-based Smoothing, the earlier observed negative correlation under affine transformation noise (Fig. 3) is tempered by the max and β-shift operations in Eq. 5, resulting in entropy behavior similar to that of a conventional neural network.

3.3 COMPUTATIONAL COST

Table 1 compares inference costs on CIFAR-10 in terms of peak memory usage and approximate FLOPs. We use the same parameters as in our previous CIFAR-10 experiments. For MC-dropout, $n = 10$ samples are processed in parallel in a single forward pass with batch size 128 · n. We include an ensemble as an additional reference with $M = 10$ models loaded simultaneously. In the table, C_{fwd} represents the total FLOPs for one forward pass, and C_{std} is the small overhead of computing the standard deviations across the logits.

MC-dropout and ensembles require multiple forward passes, resulting in higher computational costs. MC-dropout requires n times more FLOPs and 11.5 times more memory than the conventional NN, while the ensemble requires M times more FLOPs and 9.5 times more memory. In contrast, Variance-based Smoothing adds only a small overhead for variance calculations, keeping both FLOPs and memory requirements close to the conventional NN. This makes it especially suitable for real-time applications.

Table 1: Maximum memory usage (GB) and FLOPs during a single forward call on a 128-sized batch of 32 × 32 RGB images with the architecture used for the CIFAR-10 experiments.

Method	**Memory (GB)**	**FLOPs**
Conventional	0.35 ± 0.00	C_{fwd}
Temp. Scaling	0.35 ± 0.00	C_{fwd}
Ensemble ($M = 10$)	3.34 ± 0.14	$M \cdot C_{\text{fwd}}$
MC-dropout ($n = 10$)	4.04 ± 0.00	$n \cdot C_{\text{fwd}}$
Ours	0.35 ± 0.06	$C_{\text{fwd}} + C_{\text{std}}$

3.4 VARIANCE-BASED SMOOTHING AS AN EXTENSION OF ENSEMBLES

Continuing the discussion on the flexibility of ensemble distributions in large-class-count datasets (Sec. 2.3), we empirically study the behavior of these ensembles on LibriSpeech in comparison to Variance-based Smoothing as an ensemble extension. In our approach, the variance is thus computed across logits from different models rather than sub-patches. We repurpose the conventional NNs' weights from the LibriSpeech experiments (3.1) as an ensemble with $M = 10$ models. For the Variance-based Smoothing ensemble, we set $\alpha = 1$ and β to the negative 75th percentile of all $\bar{\sigma}$ values computed across the validation set.

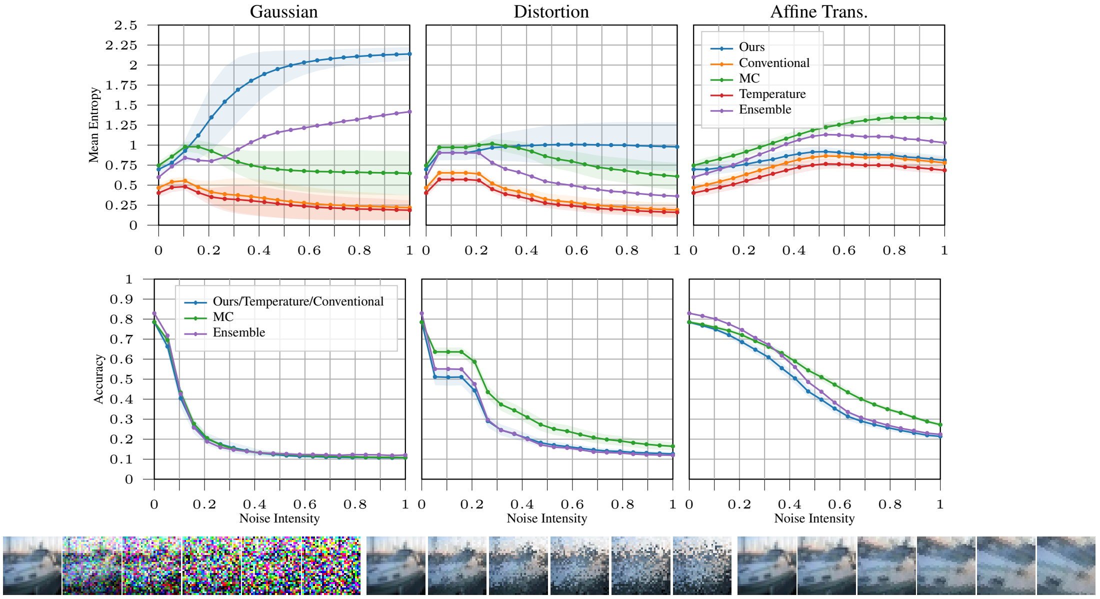

Figure 5: **CIFAR-10**: Mean entropy and their respective accuracies across different noise intensities. Our approach and Temperature Scaling have the same accuracy as conventional.

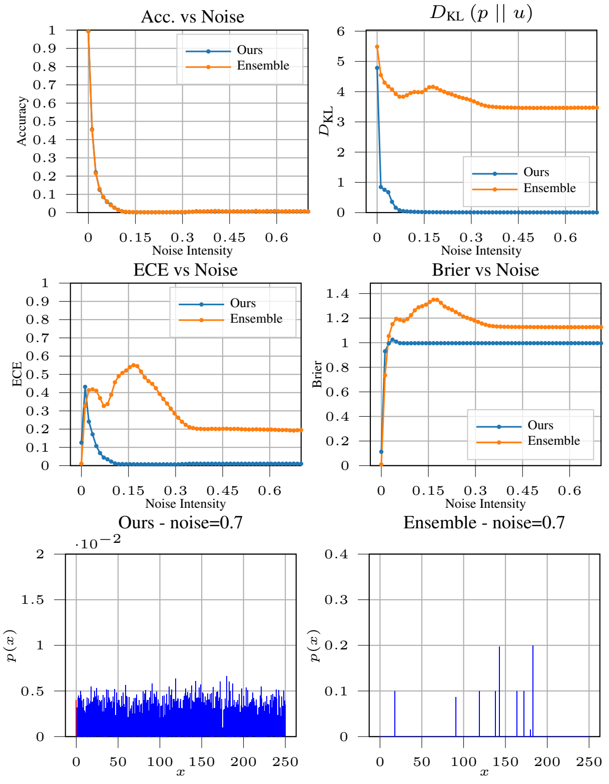

Figure 6: Comparison of ensemble flexibility on LibriSpeech under increasing Gaussian noise.

The results in Fig. 6 show that accuracy deteriorates at a similar rate for both methods as Gaussian noise increases. However, key differences appear in uncertainty estimation and calibration:

The KL-divergence with the uniform distribution starts high for both methods, indicating high confidence on the clean set. As noise increases and accuracy drops, KL-divergence decreases, as expected. However, the ensemble stagnates at KL-divergence around 3.5, failing to approach the optimal 0. In contrast, Variance-based Smoothing continues to decrease smoothly to 0, without the undesired fluctuations seen in the ensemble.

The expected calibration error (ECE), which measures the weighted average difference between predicted confidence and actual accuracy across multiple confidence bins (0 being optimal), is initially lower for ensembles due to their high-confidence predictions frequently being correct. However, as noise increases beyond a certain threshold (approximately 0.05), Variance-based Smoothing significantly outperforms ensembles, achieving near-zero calibration errors at higher noise levels.

This discrepancy is further illustrated by the two examples showing an ensemble's predicted distribution compared to ours on a distorted input with Gaussian noise = 0.7. The ensemble produces a few sharp peaks but fails to approximate the required uniform distribution over 251 classes and assigns 0 probability to the true class. Variance-based Smoothing, however, generates a near-uniform distribution while still assigning some probability to the true class label (marked in red, first column of the distribution's figure).

Additionally, the Brier score, which captures both calibration and sharpness of probabilistic predictions [DeGroot and Fienberg, 1983], shows that Variance-based Smoothing generally achieves lower values under dataset shift, indicating better overall uncertainty estimation.

These results suggest that small ensembles ($M = 10$) lack the expressiveness needed to represent a uniform distribution in high-class-count settings. Variance-based Smoothing provides a viable alternative, achieving more flexible uncertainty estimates with minimal computational overhead.

4 RELATED WORK

Ensemble-based and Efficient Single-Model UQ Approaches: Deep Ensembles [Lakshminarayanan et al., 2017] remain a gold standard for UQ due to their robust performance under dataset shift [Snoek et al., 2019]. However, they require multiple forward passes and substantial memory, limiting their practicality, especially for edge applications. Extensions like Deep Sub-Ensembles [Valdenegro-Toro, 2019] reduce computational costs by training subsets of models but still require significant memory overhead.

Several single-model approaches have been developed. MC Dropout [Gal and Ghahramani, 2016] approximates Bayesian inference via stochastic forward passes. SWA-Gaussian [Maddox et al., 2019] produce uncertainty estimates from a single model by weight averaging over multiple training checkpoints, while Spectral-normalized Neural Gaussian Processes (SNGP) [Liu et al., 2023] achieves this through distance-awareness. Other approaches, such as Simultaneous Quantile Regression (SQR) and Orthonormal Certificates (OCs) [Tagasovska and Lopez-Paz, 2019] and Evidential Deep Learning (EDL) [Sensoy et al., 2018], modify the training objective to directly predict uncertainty. Although these methods provide viable alternatives, they typically require either multiple forward passes over the full input or significant modifications during training.

Post-hoc Calibration Methods: Post-hoc methods aim to improve calibration without modifying the model or training process. Temperature scaling [Guo et al., 2017] learns a single temperature parameter on a validation set to rescale softmax outputs, thereby improving calibration. Recent extensions, such as Dirichlet calibration [Kull et al., 2019] and Bayesian Binning into Quantiles (BBQ) [Naeini et al., 2015], allow for flexible adjustment of predictive distributions. These methods rely on dataset-wide calibration parameters, whereas our approach dynamically adjusts predictions based on variance between sub-predictions, thereby also improving robustness under dataset shift.

Test-time augmentation methods for UQ [Ayhan and Berens, 2018, Jiang et al., 2022, Wu and Williamson, 2024] improve uncertainty estimates by generating multiple predictions using data augmentations such as cropping and rotation. While these approaches also rely on data redundancy, they use artificial transformations rather than naturally occurring redundancy in the data and still require multiple forward passes. Among existing methods, this is the most closely related to exploiting the structure of the input data for UQ.

5 DISCUSSION

Our main finding is that variance between sub-predictions can serve as a fast and low-cost proxy for uncertainty across different tasks and noise settings. Interestingly, our results indicate that this metric may be viable beyond tasks that strictly satisfy the informative sub-patches assumption, suggesting potential for broader applicability. When used as a post-hoc recalibration method through Variance-based Smoothing, uncertainty estimates improve significantly on both clean and noisy data. In many cases, it performs on par with, and occasionally surpasses, MC-dropout, despite being considerably less computationally expensive. We note, however, that our method is primarily effective for calibration under input perturbations rather than out-of-distribution (OOD) detection; preliminary experiments on held-out classes (see Appendix E) suggest that distinguishing OOD samples may require complementary techniques. Finally, applying Variance-based Smoothing as an extension to ensembles by using variance between model predictions improves their flexibility and leads to lower expected calibration errors in high-class-count settings.

Limitations and Future Work: While Variance-based Smoothing using variance from sub-predictions shows strong results, its effectiveness appears somewhat dataset- and noise-dependent. Future work should further investigate its performance across a wider range of datasets and noise types to better understand its limitations and generalization capabilities. Since our experiments focus solely on classification in ConvNet-based architectures, generalization to other architectures and machine learning tasks should be explored.

Our approach relies on standard deviation across logits as a measure of disagreement between sub-predictions. However, this is a relatively simple metric, and the softmax function's exponential nature causes logits to be transformed nonlinearly, which can impact variance estimation. Future research could explore alternatives that better account for this nonlinearity and potentially improve uncertainty estimates.

Lastly, Variance-based Smoothing computes variance on the model's final-layer outputs. Future work could explore whether intermediate feature activations provide additional uncertainty information. Easier-interpretable latent space regularizers, such as Denoodt et al. [2024], Burgess et al. [2018], may offer a promising starting point for studying this behavior.

References

Murat Seckin Ayhan and Philipp Berens. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. In *Medical Imaging with Deep Learning*, 2018.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loïc Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *CoRR*, abs/1804.03599, 2018. URL `http://arxiv.org/abs/1804.03599`.

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.

Fabian Denoodt, Bart de Boer, and José Oramas. Smooth infomax - towards easier post-hoc interpretability. *CoRR*, abs/2408.12936, 2024. doi: 10.48550/ARXIV. 2408.12936. URL `https://doi.org/10.48550/arXiv.2408.12936`.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org, 2016. URL `http://proceedings.mlr.press/v48/gal16.html`.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/guo17a.html`.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL `http://arxiv.org/abs/1503.02531`.

Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Wenming Jiang, Ying Zhao, Yihan Wu, and Haojia Zuo. Capturing model uncertainty with data augmentation in deep learning. In Arindam Banerjee, Zhi-Hua Zhou, Evangelos E. Papalexakis, and Matteo Riondato, editors, *Proceedings of the 2022 SIAM International Conference on Data Mining, SDM 2022, Alexandria, VA, USA, April 28-30, 2022*, pages 271–279. SIAM, 2022. doi: 10.1137/1.9781611977172.31. URL `https://doi.org/10.1137/1.9781611977172.31`.

Alex Krizhevsky and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research), 2009. URL `https://www.cs.toronto.edu/~kriz/cifar.html`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6402–6413, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html`.

M Lin. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Jeremiah Zhe Liu, Shreyas Padhy, Jie Ren, Zi Lin, Yeming Wen, Ghassen Jerfel, Zachary Nado, Jasper Snoek, Dustin Tran, and Balaji Lakshminarayanan. A simple approach to improve single-model deep uncertainty via distance-awareness. *Journal of Machine Learning Research*, 24(42):1–63, 2023. URL `https://jmlr.org/papers/v24/22-0479.html`.

Sindy Löwe, Peter O'Connor, and Bastiaan S. Veeling. Putting an end to end-to-end: Gradient-isolated learning of representations. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3033–3045, 2019. URL `https://proceedings.`

neurips.cc/paper/2019/hash/851300ee84c2b80ed40f51ed26d866fc-Abstract.html.

Wesley J. Maddox, Pavel Izmailov, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13132–13143, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/118921efba23fc329e6560b27861f0c2-Abstract.html.

Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2901–2907. AAAI Press, 2015. doi: 10.1609/AAAI.V29I1.9602. URL https://doi.org/10.1609/aaai.v29i1.9602.

Thayheng Nhem, Fabian Denoodt, Maarten Weyn, Michael Peeters, José Oramas, and Rafael Berkvens. Label-efficient learning for radio frequency fingerprint identification. In *IEEE Wireless Communications and Networking Conference, WCNC 2025*, 2025. To appear.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 5206–5210. IEEE, 2015. doi: 10.1109/ICASSP.2015.7178964. URL https://doi.org/10.1109/ICASSP.2015.7178964.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3967–3976. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00409. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Park_Relational_Knowledge_Distillation_CVPR_2019_paper.html.

Murat Sensoy, Lance M. Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3183–3193, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/a981f2b708044d6fb4a71a1463242520-Abstract.html.

Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua V. Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13969–13980, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8558cb408c1d76621371888657d2eb1d-Abstract.html.

Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/73c03186765e199c116224b68adc5fa0-Paper.pdf.

Matias Valdenegro-Toro. Deep sub-ensembles for fast uncertainty estimation in image classification. *CoRR*, abs/1910.08168, 2019. URL http://arxiv.org/abs/1910.08168.

Luhuan Wu and Sinead A. Williamson. Posterior uncertainty quantification in neural networks using data augmentation. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 3376–3384. PMLR, 2024. URL https://proceedings.mlr.press/v238/wu24e.html.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network, 2015. URL https://arxiv.org/abs/1505.00853.

Efficient Post-Hoc Uncertainty Calibration via Variance-Based Smoothing (Supplementary Material)

A EXPERIMENTAL SETUP

All single-model experiments are repeated 10 times with unique random seeds.

A.1 RADIO

We follow Nhem et al. [2025] unless stated otherwise. The model is trained in a fully supervised manner using the Adam optimizer for 30 epochs with a learning rate of 1×10^{-3} and a batch size of 32. Training is performed using cross-entropy loss.

The model architecture, shown in Table 2, is identical to Nhem et al. [2025], except that the GRU layer is omitted for simplicity. Batch normalization is applied after each convolution layer, followed by a ReLU activation, except for the final layer, where neither batch normalization nor ReLU is applied.

The dataset follows the train-validation-test split used by Nhem et al. [2025], with 70,000 training examples (70%), 9,900 validation examples (10%), and 20,100 test examples (20%). To improve consistency between sub-predictions, the first 500 samples of each example are removed. As a result, predictions are made on 1580×2 inputs, resulting in a 79×20 tensor before the final average pooling layer. The model has a total downscaling factor of 20.

Table 2: Architecture for the Radio experiments. ^{a}Variance is computed across logits from this layer.

Layer	Output	Kernel	Stride	Padding
Input	1580×2			
Conv	315×512	10	5	2
Conv	78×512	8	4	2
Conv	79×512	4	1	2
Conv	80×512	4	1	2
Conv	79×512	4	1	1
Conv	79×20^{a}	1	1	-
Avg. pool	1×20	-	-	-

A.2 LIBRISPEECH

The protocol follows that of the fully supervised LibriSpeech models described in Löwe et al. [2019], Denoodt et al. [2024]. The architecture, shown in Table 3, is the same, except that the GRU layer is removed, and batch normalization is added after each convolution layer, except for the final one.

The model is trained using the Adam optimizer with a learning rate of 2×10^{-4} for 1,000 epochs with a batch size of 8.

The dataset consists of spoken audio sampled at 16 kHz. Since Löwe et al. [2019] does not explicitly provide a validation set, we split the test in half with 50% for validation and 50% for testing, resulting in 22,830 training examples (80%), 2,854 validation examples (10%), and 2,854 test examples (10%). As examples contain variable-length audio, random crops of length 20,480 (1.28s) are extracted for training and evaluation.

Table 3: Architecture for the LibriSpeech experiments. [a]Variance is computed across logits from this layer.

Layer	**Output**	**Kernel**	**Stride**	**Padding**
Input	20480×1			
Conv	4095×512	10	5	2
Conv	1023×512	8	4	2
Conv	512×512	4	2	2
Conv	257×512	4	2	2
Conv	128×512	4	2	1
Conv	128×251^a	1	1	-
Average pool	1×251	-	-	-

A.3 CIFAR-10

We use the architecture proposed by Xu et al. [2015], which is based on the Network-In-Network (NiN) architecture [Lin, 2013] and provides relatively strong performance on CIFAR-10. The architecture is presented in Table 4.

The model is trained using the Adam optimizer with a learning rate of 1×10^{-3} for 250 epochs with a batch size of 128 and a weight decay of 5×10^{-4}. Note that for our MC-dropout implementations, the already present dropout layers are omitted and an additional dropout layer is added after every ReLu layer.

Table 4: Architecture for the CIFAR-10 experiments. [a]Variance is computed across logits from this layer.

Layer	**Output**	**Kernel**	**Stride**	**Padding**
Input	$32 \times 32 \times 3$			
Conv	$32 \times 32 \times 192$	5×5	1	2
Conv	$32 \times 32 \times 160$	1×1	1	0
Conv	$32 \times 32 \times 96$	1×1	1	0
Max Pool	$16 \times 16 \times 96$	3×3	2	1
Dropout (0.5)	-	-	-	-
Conv	$16 \times 16 \times 192$	5×5	1	2
Conv	$16 \times 16 \times 192$	1×1	1	0
Conv	$16 \times 16 \times 192$	1×1	1	0
Avg. Pool	$8 \times 8 \times 192$	3×3	2	1
Dropout (0.5)	-	-	-	-
Conv	$8 \times 8 \times 192$	3×3	1	1
Conv	$8 \times 8 \times 192$	1×1	1	0
Conv	$8 \times 8 \times 10^a$	1×1	1	0
Avg. Pool	$1 \times 1 \times 10$	8×8	-	-

A.4 NOISE TYPES

Across the three datasets, inputs are incrementally distorted using different noise types with the degree of distortion controlled by $\lambda \in [0, 1]$. For Radio and LibriSpeech we apply Gaussian or Speckle noise, while for CIFAR-10 we apply either Gaussian noise, affine transformations, or elastic distortions.

Gaussian noise is introduced as $\mathbf{x}_{\text{noise}} := \mathbf{x} + \lambda\epsilon$, while Speckle noise is applied as $\mathbf{x}_{\text{noise}} := \mathbf{x} + \lambda(\mathbf{x} \odot \epsilon)$, where $\odot$ denotes element-wise multiplication. In both cases, ϵ is a noise vector $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ of the same shape as the input $\mathbf{x}$ is sampled.

In the case of an affine transformation, an image $\mathbf{X}$ of size $H \times W$ is modified through a rotation by $\theta = \lambda \cdot 30°$, a shear by $s = \lambda \cdot 10°$, and an isotropic scaling factor $\gamma = 1 + \lambda$. To prevent cropping artificats, the image is first padded by $p = 0.2 \max(H, W)$ before applying the transformation, which is parameterized by the matrix

$$A = \begin{bmatrix} \gamma \cos\theta & -\gamma \sin\theta + s \\ \gamma \sin\theta & \gamma \cos\theta + s \end{bmatrix}. \tag{8}$$

Finally, the image is center-cropped back to its original dimensions to ensure that the structured perturbations induced by λ are smoothly incorporated while maintaining spatial consistency.

When elastic distortion is applied, pixel coordinates are perturbed based on randomly generated displacement fields. Given an image $\mathbf{X}$, displacement fields Δx and Δy are sampled independently from a uniform distribution $\mathcal{U}(-\lambda \cdot 5, \lambda \cdot 5)$. Each pixel coordinates (x, y) is displaced to

$$x' = \text{clamp}(x + \Delta x, 0, W - 1), \quad y' = \text{clamp}(y + \Delta y, 0, H - 1), \tag{9}$$

where the clamp function ensures that all coordinates remain within valid image bounds. The deformed image $\mathbf{X}'$ is then obtained by mapping each pixel to its new location using nearest-neighbor interpolation, introducing local distortions that increase in magnitude as λ approaches 1 while preserving overall spatial coherence.

Visual examples of the noise types applied to an image are shown in Fig. 5. We also provide code implementations for all noise types in the GitHub repository.

A.5 BASELINES

We compare our method against MC-dropout and Temperature Scaling. MC-dropout is trained with the same hyperparameters as the conventional NN, except that dropout ($p = 0.5$) is applied after every ReLU layer. During evaluation, dropout is maintained, and inference is performed on batches of size ($n \cdot$ Orig. Batch Size), with $n = 10$. For Temperature Scaling, we reuse the conventional NN's weights and continue training the temperature parameter on top of these frozen weights. This scalar is initialized to 1 and optimized for 10 epochs using the Adam optimizer, with a learning rate of 0.01, a batch size of 64, and a Cross-Entropy loss.

B RELIABILITY DIAGRAMS AND CONFIDENCE COUNTS

B.1 RADIO - CLEAN DATA

Fig. 7 displays the individual reliability diagrams and their respective confidence counts for 1) clean test inputs, 2) Gaussian perturbed test inputs with noise intensity of 0.2, and finally 3) Gaussian perturbed test inputs with noise intensity of 0.5.

B.2 LIBRISPEECH - CLEAN DATA

Fig. 8 displays the individual reliability diagrams and their respective confidence counts for 1) clean test inputs, 2) Gaussian perturbed test inputs with noise intensity of 0.2, and finally 3) Gaussian perturbed test inputs with noise intensity of 0.5.

B.3 CIFAR-10 - CLEAN DATA

Fig. 9 displays the individual reliability diagrams and their respective confidence counts for 1) clean test inputs, 2) Gaussian perturbed test inputs with noise intensity of 0.2, and finally 3) Gaussian perturbed test inputs with noise intensity of 0.5.

C PSEUDOCODE

Algorithm 1 describes a conventional NN's forward pass and algorithm 2 describes our proposed Variance-based Smoothing approach.

Algorithm 1 Forward Pass with a Conventional NN

```
Input: Input sample $\mathbf{x}$                                  ▷ Shape: $(C, L)$
Get pre-pooling logits $\mathbf{z}_{1:L'} \leftarrow \text{CNN}(\mathbf{x})$                ▷ Shape: $(L', K)$
Get final logits: $\mathbf{z} \leftarrow \frac{1}{L'} \sum_{l=1}^{L'} \mathbf{z}_l$      ▷ Corresponds to $p_{\text{logit}}(y \mid \mathbf{x})$ in Eq. 3, Shape: $(K, )$
Apply softmax: $p(y|\mathbf{x}) = \text{Softmax}(\mathbf{z})$                ▷ No temperature scaling
Output: Raw probabilities $p(y|\mathbf{x})$                       ▷ Shape: $(K, )$
```

Algorithm 2 Forward Pass with Variance-Based Smoothing

```
Input: Input sample $\mathbf{x}$                                  ▷ Shape: $(C, L)$
Get pre-pooling logits $\mathbf{z}_{1:L'} \leftarrow \text{CNN}(\mathbf{x})$                ▷ Shape: $(L', K)$
Get $p_{\text{logit}}(y|\mathbf{x}_t) \leftarrow \text{AvgPool}(\mathbf{z}_{1:L'}, \text{kernel-size}=s, \text{stride}=1)$   ▷ Merge neighbors; discussed in Sec. 3.1, Shape: $(T, K)$
Compute per-class statistics:
for $k \leftarrow 1$ to $K$ do
    $\mu_k \leftarrow \frac{1}{T} \sum_{t=1}^{T} [p_{\text{logit}}(y|\mathbf{x}_t)]_k$                  ▷ Class means
    $\sigma_k \leftarrow \sqrt{\frac{1}{T-1} \sum_{t=1}^{T} ([p_{\text{logit}}(y|\mathbf{x}_t)]_k - \mu_k)^2}$    ▷ From Eq. 4
end for
Aggregate variance: $\bar{\sigma} \leftarrow \frac{1}{K} \sum_{k=1}^{K} \sigma_k$               ▷ From Eq. 5
Compute temperature: $\tilde{\sigma} \leftarrow \max(\alpha(\bar{\sigma} + \beta), 1)$          ▷ From Eq. 5
Get final logits: $\mathbf{z} \leftarrow \frac{1}{L'} \sum_{l=1}^{L'} z_l$      ▷ Corresponds to $p_{\text{logit}}(y \mid \mathbf{x})$ in Eq. 3, Shape: $(K, )$
Apply scaled softmax: $p(y|\mathbf{x}) = \text{Softmax}(\mathbf{z}/\tilde{\sigma})$          ▷ From Eq. 3
Output: Calibrated probabilities $p(y|\mathbf{x})$                ▷ Shape: $(K, )$
```

D ABLATION STUDIES ON THE PATCH-SIZE, α AND β

Following the input splitting and sub-patch processing discussion in Sec. 3.1, we conduct an ablation study on the patch-size (or kernel-size) (see Fig. 10) and the hyperparameters α (see Fig. 11) and β (see Fig. 12).

E OOD EXPERIMENTS ON HELD-OUT CLASSES

Fig. 13 depicts the distribution of predictive uncertainty (measured through entropy) on predictions of seen versus unseen classes. The plots depict the result of 10 different models, each time trained on a set where class [i] is omitted. We find that Variance-based Smoothing's distributions on seen and unseen classes are relatively similar, indicating high confidence in predictions of unseen classes. Temperature Scaling behaves similarly, while MC-dropout can better distinguish the unseen classes from the seen ones.

[Results for LibriSpeech and CIFAR-10 will be placed here.]

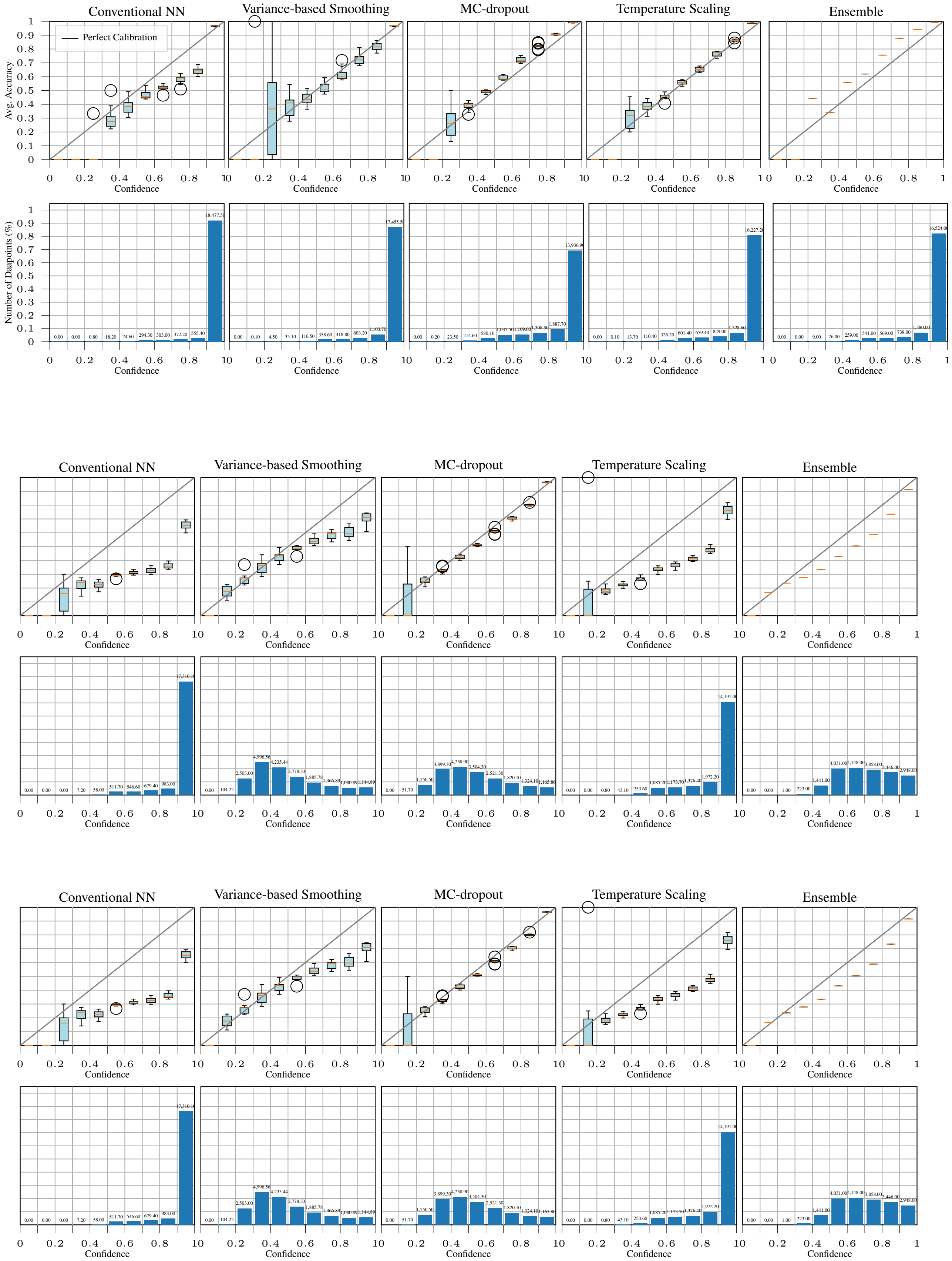

Figure 7: **Radio**: Individual reliability diagrams and their frequency per bin (averaged over 10 runs) on clean test data (first 2 rows), Gaussian noise with intensity 0.2 (middle 2 rows), and Gaussian noise with intensity 0.5 (last 2 rows).

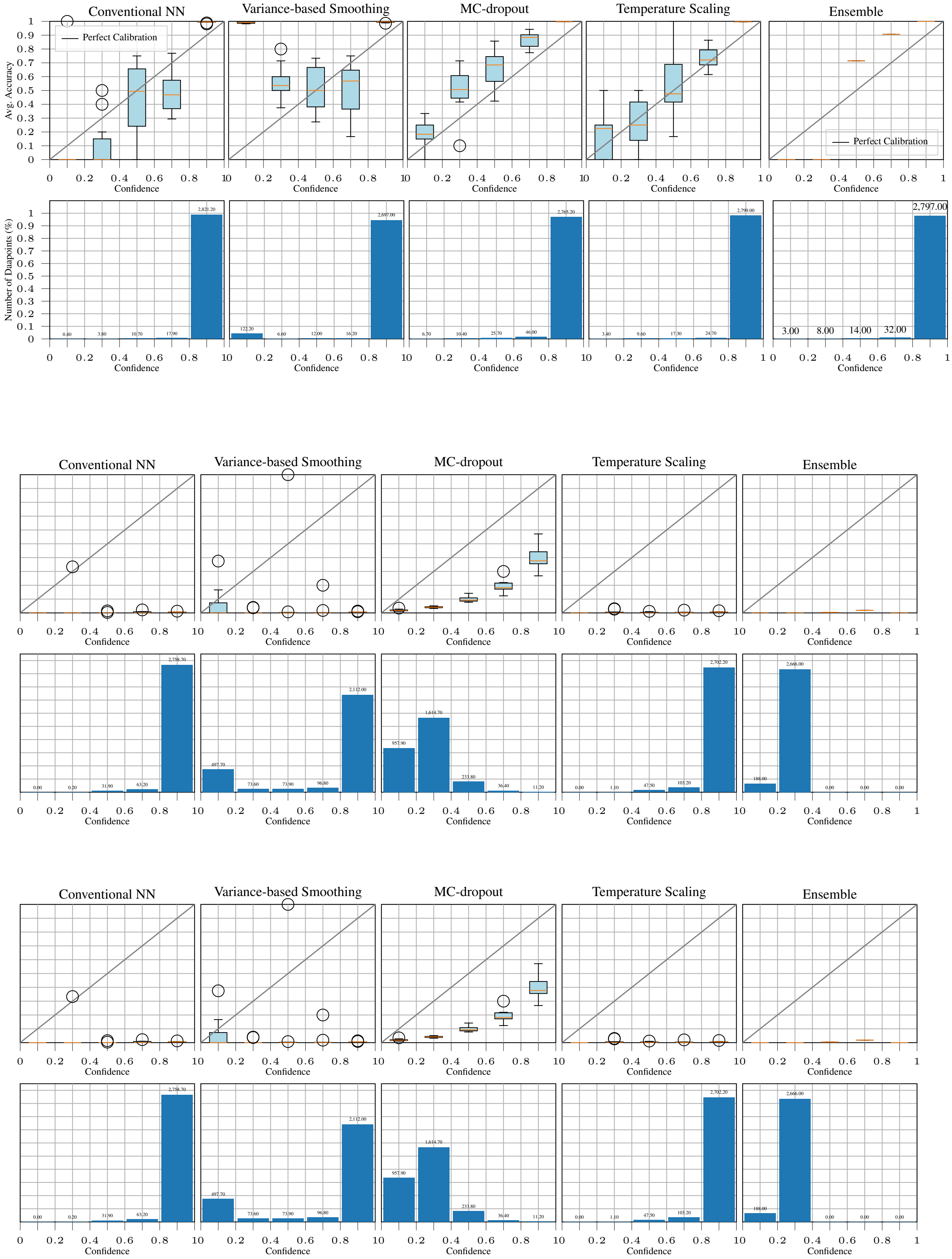

Figure 8: **LibriSpeech**: Individual reliability diagrams and their frequency per bin (averaged over 10 runs) on clean test data (first 2 rows), Gaussian noise with intensity 0.2 (middle 2 rows), and Gaussian noise with intensity 0.5 (last 2 rows).

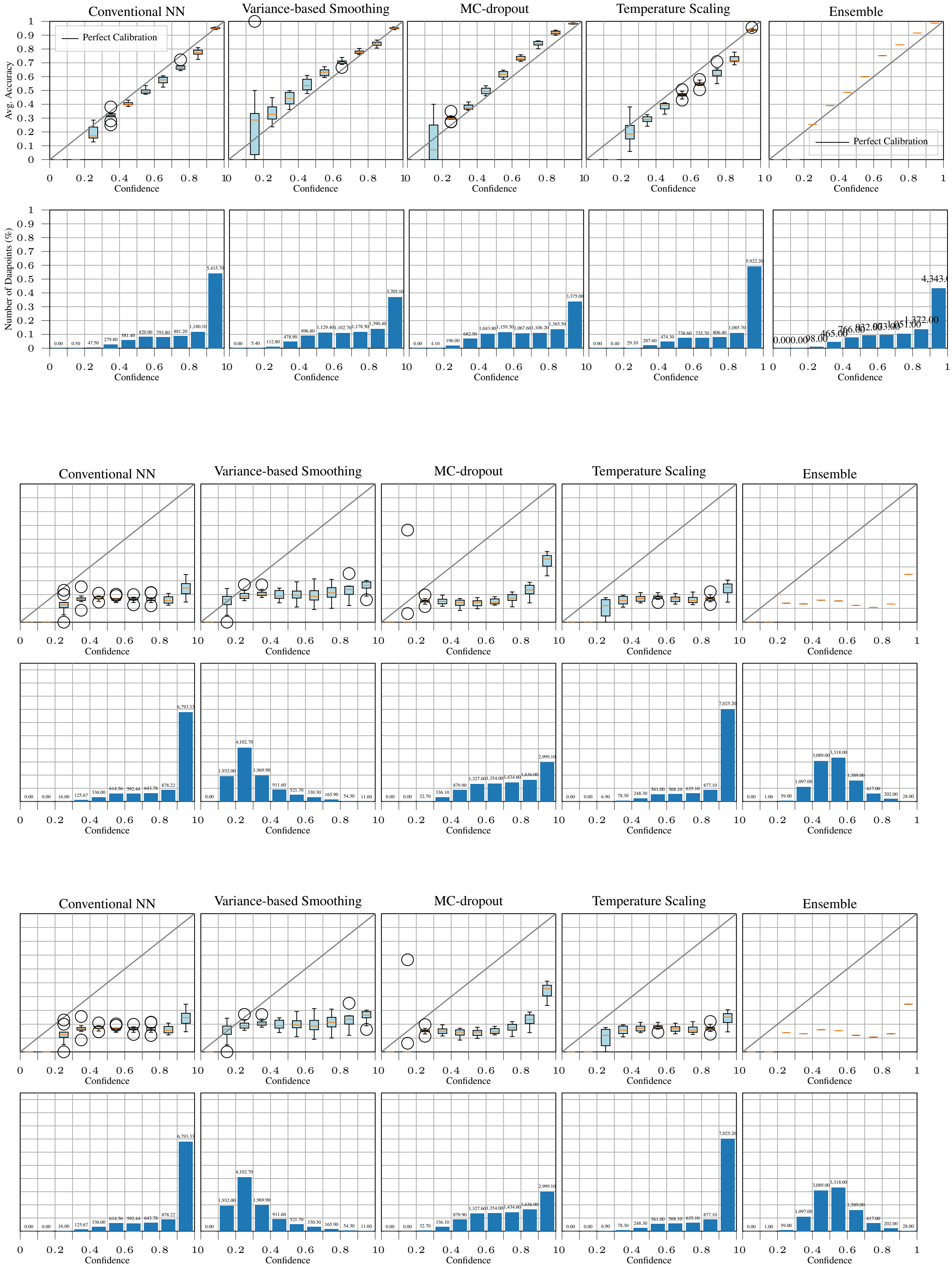

Figure 9: **CIFAR-10**: Individual reliability diagrams and their frequency per bin (averaged over 10 runs) on clean test data (first 2 rows), Gaussian noise with intensity 0.2 (middle 2 rows), and Gaussian noise with intensity 0.5 (last 2 rows).

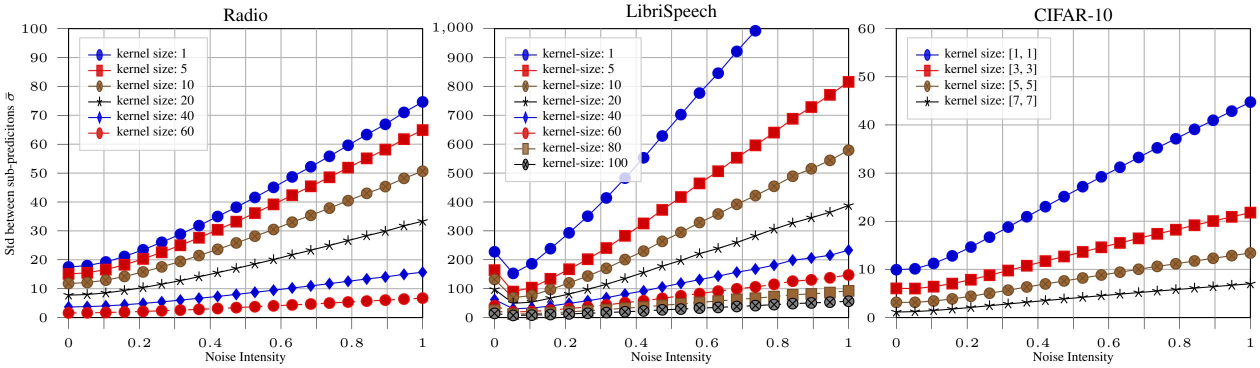

Figure 10: Ablation study on kernel size for Radio, LibriSpeech, and CIFAR-10 datasets.

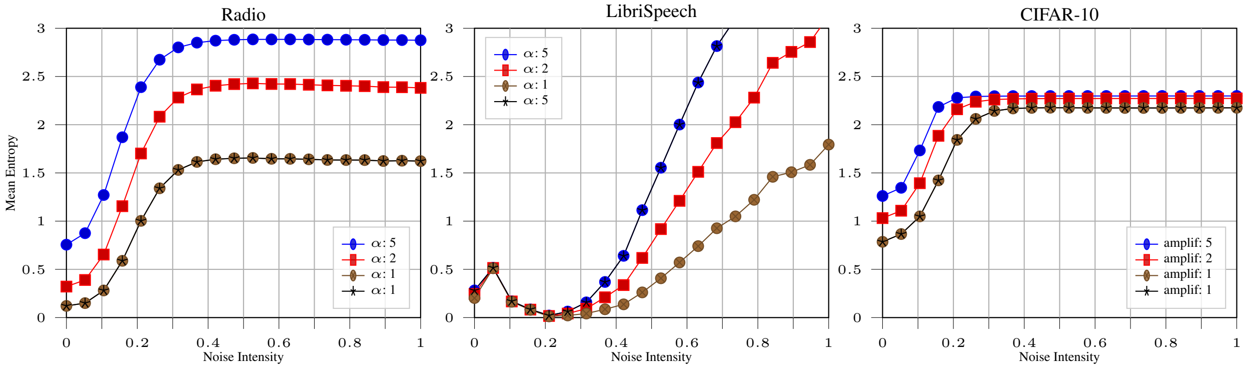

Figure 11: Ablation study on α amplification for Radio, LibriSpeech, and CIFAR-10 datasets.

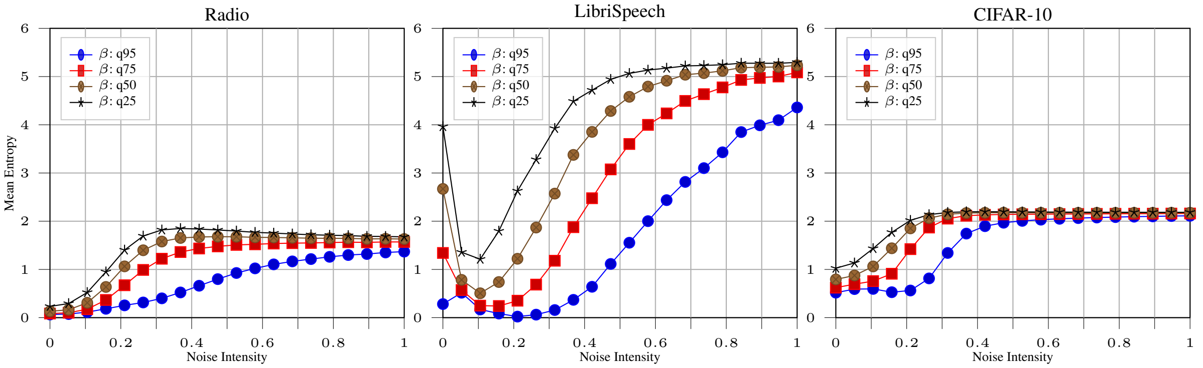

Figure 12: Ablation study on β for Radio, LibriSpeech, and CIFAR-10 datasets.

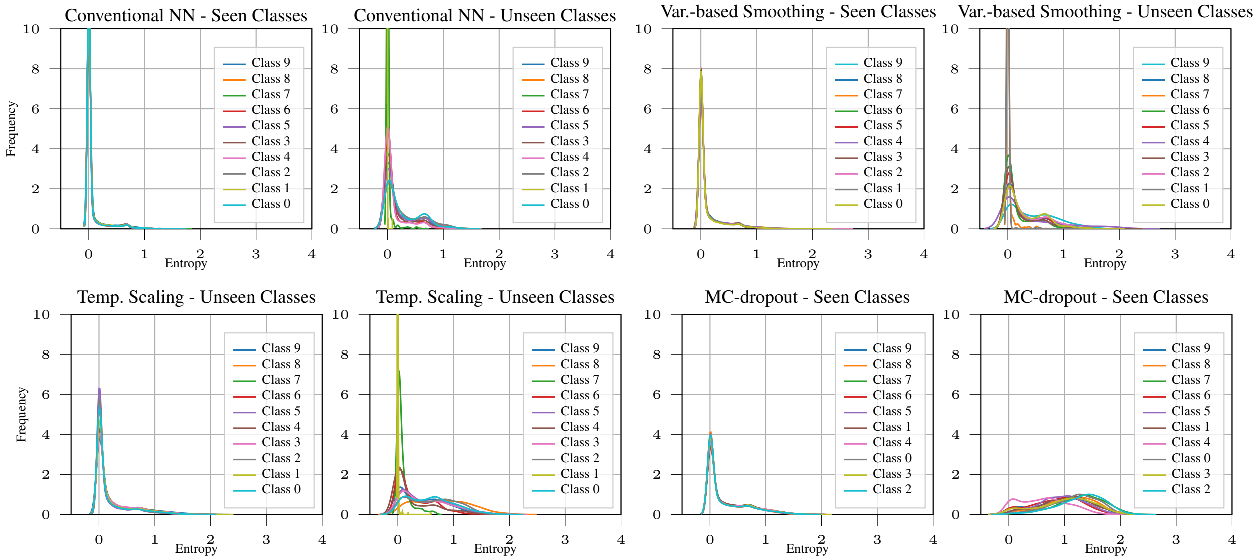

Figure 13: Distribution over predictive entropy on seen and unseen classes for the Radio dataset. Models are trained on a training set where class [i] is omitted. Afterward, predictions are made on inputs that either belong to the class (Seen classes) or not (Unseen class).