Where to Drop: Tuning Monte Carlo Dropout for Uncertainty Calibration in Image Classification

Lina Benyamina
ENSAE Paris, Institut Polytechnique de Paris

Emilien Jemelen
HeKa team (Inria, Inserm, Université Paris Cité)

Abstract

Calibrated uncertainty is essential for deploying deep neural networks in high-stakes settings such as medical diagnosis. Monte Carlo Dropout (MC Dropout) provides a practical Bayesian approximation within a single architecture, yet key choices—dropout probability and mask placement—are typically heuristic and can produce uncertainty scores that poorly track error rates. We therefore perform a systematic grid search over MC Dropout hyperparameters and assess calibration via the monotonic relationship between accuracy and uncertainty, using accuracy–uncertainty curves with monotonicity-aware evaluation. On CIFAR-10, performance varies widely across configurations, but applying dropout in the penultimate layer consistently yields the most monotonic, actionable degradation as uncertainty increases. We validate this "penultimate-layer rule" on mammography triage for breast cancer screening, where calibrated uncertainty is crucial for safe deferral and workload allocation. Code and reproducibility artifacts are released at `https://github.com/linabny/MonteCarlo_Dropout`.

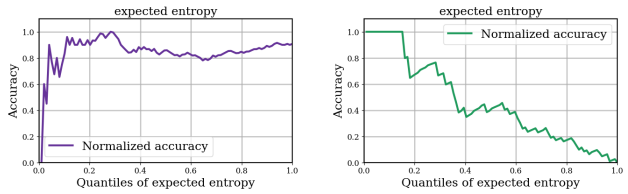
1 Introduction

In high-stakes clinical settings, predictive systems must be able to recognize when their outputs may be unreliable. In cancer detection, for example, the costs of misclassification are asymmetric: false positives can cause unnecessary psychological distress and increase healthcare costs, while false negatives can be fatal by delaying treatment. Standard deep neural networks, despite their high accuracy, often behave as overconfident "black boxes" that fail to indicate when they are likely to be wrong (Lakshminarayanan et al., 2017). To be safely deployed in critical decision-making—such as disease diagnosis, fraud detection, or autonomous driving—these models must provide well-calibrated uncertainty: a trustworthy confidence measure that allows the system to defer ambiguous cases to a qualified expert for review.

To estimate predictive uncertainty, a range of confidence metrics has been proposed, from simple heuristics such as the maximum softmax probability (Hendrycks and Gimpel, 2017) and learned confidence scores (Corbière et al., 2019; DeVries and Taylor, 2018) to distance-based measures in feature space (Wang et al., 2023; van Amersfoort et al., 2020). More advanced methods, such as Deep Ensembles, can deliver high-quality calibration by training multiple models, but at the cost of substantially increased computation (Lakshminarayanan et al., 2017). Among these approaches, Monte Carlo Dropout (MC Dropout) has become a widely used baseline thanks to its theoretical grounding and its ability to approximate Bayesian inference within a single architecture, without changing the training procedure (Gal and Ghahramani, 2016). In medical imaging in particular, MC Dropout has been used extensively to quantify uncertainty in classification, segmentation, and reconstruction tasks (Villanueva Galapon Jr et al., 2024; Medina and co authors, 2022; Maleki Sadr et al., 2022; Nair et al., 2020).

Related Work. The theoretical basis for uncertainty estimation with MC Dropout in deep learning was established by Gal and Ghahramani (2016), who interpreted dropout as a practical approximation to Deep Gaussian Processes (Damianou and Lawrence, 2013). The quality of MC Dropout uncertainty estimates is commonly assessed using accuracy–uncertainty curves, which evaluate how reliably uncertainty scores reflect performance; for a given task, these scores are expected to correlate monotonically

Workshop "Towards Trustworthy Predictions: Theory and Applications of Calibration for Modern AI" at AISTATS 2026, Tangier, Morocco.



(a) Poorly calibrated accuracy curve.

(b) Well-calibrated accuracy curve.

Figure 1: Comparison of calibration quality for two MC Dropout configurations in classification task. Uncertainty is measured by expected entropy, which should be negatively correlated with accuracy. Figure 1a illustrates a poorly calibrated configuration, where higher uncertainty does not correspond to lower accuracy. In contrast, Figure 1b shows a well-calibrated configuration in which accuracy decreases smoothly as uncertainty increases. Dataset: CIFAR-10.

with the model's accuracy (Gu et al., 2024; Valiuddin et al., 2024). Iterative isotonic regression via the Pool Adjacent Violator Algorithm (PAVA) (Jégou, 2012; Ayer et al., 1955) and monotone rearrangement (Chernozhukov et al., 2009) provide principled tools to quantify deviations from this ideal strictly decreasing relationship.

However, a critical gap remains: in the current literature, MC Dropout configurations are largely heuristic, often chosen through trial and error. Practitioners typically select dropout probabilities and mask placements without systematic justification, relying on guesswork to obtain well-calibrated uncertainty estimates. As illustrated in Figure 1, arbitrary configurations can produce inconsistent behavior, where uncertainty fails to track error rates (Figure 1a). Conversely, some configurations may ensure that higher uncertainty reliably signals degraded performance (Figure 1b) (see Supplementary A.2.2 for details). To our knowledge, no empirical study has examined how MC Dropout configuration affects the monotonic relationship between convolutional network (CNN) performance and uncertainty. We address this gap.

Our contributions are:

1. We propose a systematic grid search over MC Dropout hyperparameters to identify configurations that maximize uncertainty calibration, and instantiate it on two imaging classification tasks.

2. We derive empirical, metric- and dataset-specific guidelines for obtaining well-calibrated uncertainty, and show that applying dropout to the penultimate layer consistently yields the best calibration.

2 Problem Setting: Monte Carlo Dropout theory in Classification, Uncertainty Metrics and Calibration Assessment

Classification Setting. We consider the standard classification setting. Let $\mathbb{P}$ denote the distribution of random pairs $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y} := \{1, \dots, C\}$, $C \in \mathbb{N}^*$, X denotes the features and Y the label. A classifier is any function $f : \mathcal{X} \to \mathcal{Y}$. In particular we consider f_W a network with per-layer weights $W = \{W_1, \dots, W_H\}$ defined below.

Bayesian Formalization of the Network. Following the theoretical framework of Gal and Ghahramani (2016), for each layer h we define the (random) weight matrix as

$$W_h \;=\; M_h \,\mathrm{diag}(z_{h,1}, \dots, z_{h,K_{h-1}}), \qquad z_{h,j} \sim \mathrm{Bernoulli}(p_h), \tag{1}$$

where $M_h \in \mathbb{R}^{K_h \times K_{h-1}}$ is the deterministic weight matrix, K_{h-1} denotes the width of the previous layer, and $p_h \in (0, 1]$ is the keep probability (i.e., one minus the dropout probability).

A dropout configuration $\theta = (\{M_h\}_{h=1}^H, \{p_h\}_{h=1}^H)$ induces a variational distribution $q_\theta(W)$ over the network weights. This distribution defines the Bayesian predictive distribution

$$p_\theta(c \mid x) \;=\; \mathbb{E}_{W \sim q_\theta}[\mathrm{softmax}(f_W(x))_c]\,. \tag{2}$$

We estimate the predictive probability vector by averaging over T stochastic forward passes:

$$\hat{p}_\theta(c \mid x) \;=\; \frac{1}{T}\sum_{t=1}^{T} p^{(t)}(c \mid x), \qquad c \in \{1, \dots, C\}, \tag{3}$$

where $p^{(t)}(c \mid x) = \mathrm{softmax}(f_{W^{(t)}}(x))_c$ and $W^{(t)} \overset{\text{iid}}{\sim} q_\theta$. The predicted label is then

$$\hat{y}_\theta(x) \;=\; \arg\max_{c \in \{1,\dots,C\}} \hat{p}_\theta(c \mid x). \tag{4}$$

Our aim is to evaluate different dropout configurations by varying the probabilities $\{p_h\}$ and the layers in which dropout is applied (Table 1), and to measure the impact on calibration quality.

Uncertainty Metrics for Classification. We define uncertainty metrics as functionals $\Psi(\cdot)$ of the stochastic predictions:

$$U_\theta(x) \;=\; \Psi\Big(\big\{\mathrm{softmax}(f_{W^{(t)}}(x))\big\}_{t=1}^{T}\Big). \tag{5}$$

We compute five such metrics (see Supplementary A.1 for explicit formulas):

- **Variation of the predicted class**: measures how the probability assigned to the class predicted in the reference pass varies across stochastic passes.
- **Variation of the maximum probability**: captures the stability of the model's peak confidence across passes.
- **Predictive entropy**: quantifies overall uncertainty using Shannon entropy (Shannon, 1948).
- **Expected entropy**: estimates data-inherent noise (aleatoric uncertainty) by averaging the entropy of each stochastic prediction.
- **BALD**: estimates model uncertainty (epistemic uncertainty) by computing the mutual information between predictions and the model posterior.

Calibration Evaluation. Finally, letting $F_\theta(u) := \mathbb{P}\big(U_\theta(X) \le u\big)$ denote the CDF of the uncertainty metric, we define the accuracy–uncertainty curve as the mapping $\tau \in [0,1] \mapsto \mathrm{Acc}_\theta(\tau)$, where

$$\mathrm{Acc}_\theta(\tau) = \mathbb{P}\big(\hat{y}_\theta(X) = Y \,\big|\, U_\theta(X) \le F_\theta^{-1}(\tau)\big)\,. \quad (6)$$

We estimate $\mathrm{Acc}_\theta(\tau)$ empirically by

$$\mathrm{Accuracy}(\tau) = \frac{|\mathcal{C}_\tau|}{|\mathcal{S}_\tau|}, \quad (7)$$

where $\mathcal{S}_\tau$ contains samples with uncertainty below the threshold τ and $\mathcal{C}_\tau$ is the correctly classified subset. We construct accuracy-vs-uncertainty curves by partitioning test samples into $N = 100$ quantiles, hereafter referred to as calibration curves.

We formalize empirical calibration in classification by introducing Hypothesis 1, consistent with the definition used in prior work (Gu et al., 2024).

Hypothesis 1 (Accuracy monotonicity) *Let Accuracy$(\cdot)$ be defined as in Eq. 7. Then, for any $(\tau_1, \tau_2) \in [0,1]^2$,*

$$\tau_1 \le \tau_2 \iff \textit{Accuracy}(\tau_1) \ge \textit{Accuracy}(\tau_2).$$

A well-calibrated MC Dropout configuration should minimize violations of Hypothesis 1. To quantify these violations, we measure the discrepancy between the empirical accuracy–uncertainty curve and an ideal monotonic trend, using antitonic regression (via PAVA) (Jégou, 2012) and monotone rearrangement (Chernozhukov et al., 2009) (see Supplementary A.2 for details).

We define our key calibration score, the monotonicity penalty, as the area between the raw curve $a(x)$ and its monotonic version $c(x)$:

$$\text{Penalty} = \int_0^1 |a(x) - c(x)|\, dx. \quad (8)$$

A lower penalty indicates better calibration, showing that model uncertainty is inherently aligned with empirical performance.

3 Empirical Experiments Setting

To investigate the impact of MC Dropout configurations on calibration, we developed a framework to incorporate dropout layers into any pre-trained architectures, and used it in our MC Dropout configuration grid search (summarized in Table 1). In all our experiments we set $T = 100$ (number of MC forward passes). Computation details—including models training and code reproducibility, are available in Supplementary B

Table 1: Summary of grid search parameters used in both CIFAR-10 and DDSM classification tasks.

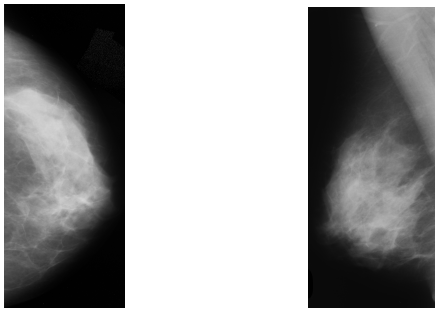
Parameter	Value
Targeted Layers	All combinations (input → penultimate)
Probabilities	$p_h \in \{0.1, 0.2, \ldots, 0.9\}$
Placement	Relative to activation function (Before/After)

Datasets and Models. We ran the grid search on two tasks. For discovery, we used CIFAR-10 (Krizhevsky, 2009), training a custom five-layer CNN with a 10-class fully connected last layer. We then validated on Mini-DDSM (Cheddad, 2020), a medical mammography dataset binarized for abnormality detection (normal vs. abnormal) following the ACR BI-RADS reference system (Destounis et al., 2025), reflecting a clinically relevant mammogram triage task (Figure 2). For this task, we used an EffNet model (Freeman et al., 2018) adapted to grayscale images.

Mammogram preprocessing (artifact removal, breast tissue isolation, augmentation) is detailed in Supplementary B.2.

4 Results: Impact of Parameters on Calibration

We present the main results of the impact of MC Dropout configurations on calibration across metrics and monotonic corrections for both tasks. Additional results are available in Supplementary C.



(a) Normal (BiRads= 1) (b) Abnormal (BiRads> 1)

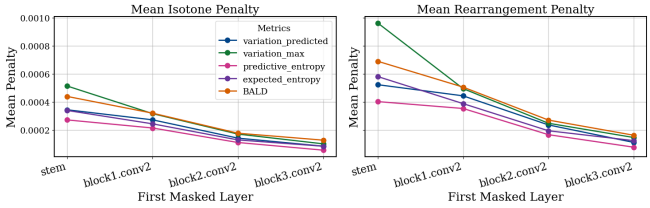
Figure 2: Figure 2a displays a normal mammogram in cranio-caudal view, and Figure 2b shows an abnormal (benign or malignant tumor in the image) mammogram in mediolateral-oblique view. Dataset: Mini-DDSM.

Dropout Probability. The impact of higher dropout probabilities is metric-dependent. On CIFAR-10, increasing the dropout probability severely degrades the calibration of predictive entropy (rearrangement penalty spikes from 0.00015 to 0.0012), but consistently improves the calibration of the variation of the predicted class, where the penalty drops from 0.13 to 0.03 (see Supplementary Figure 7).

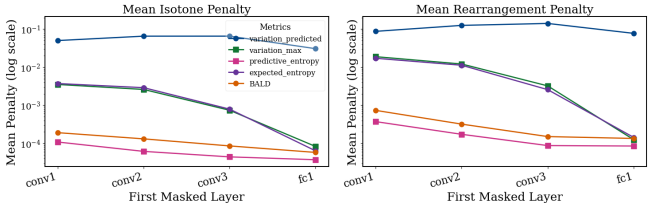
Placement and Number of Layers. No statistically significant impact was observed regarding dropout placement (before or after activation layers) or the total number of masked layers. Across all tested configurations, the standard deviation of monotonicity penalty (defined in Eq. 8) remained high, and no specific position relative to activations emerged as a primary determinant of calibration.

Dropout Depth. The depth of the last masked layer emerged as a key calibration factor. On CIFAR-10, results are metric-dependent: the rearrangement penalty for the variation of the predicted class and for the variation of the maximum probability is highest when masking is only at `conv1` (the first layer; peak values are 0.11 and 0.04, respectively), while other metrics show different sensitivities (Supplementary Figure 12). In contrast, for DDSM, calibration is optimal when dropout is confined to the early layers; the penalties then increase monotonically as the last masked layer is positioned deeper in the architecture.

Penultimate Layer Rule. Beyond these early-layer effects, our main finding is that applying dropout only to the penultimate layer (the first fully connected layer in the CIFAR-10 CNN, or the second convolutional layer in the DDSM EffNet) consistently yields the best calibration. Compared with multi-layer masking, this configuration reduces mean penalties by half, reaching a minimum rearrangement penalty for the variation of the predicted class of about 0.079 on CIFAR-10 and 0.0001 on DDSM (see Figure 3).



(a) Monotonicity penalty vs first layer masked. Dataset: DDSM (mammograms).



(b) Monotonicity penalty vs first layer masked (log scale on y-axis). Dataset: CIFAR-10.

Figure 3: Calibration (measured as monotonicity violation; lower is better) as a function of the first masked layer in the network. On the x-axis, layers are ordered from input to output. Masking only the penultimate layer consistently yields the smallest monotonicity violations, and thus the best calibration.

5 Concluding Remarks

We provide an initial empirical study of how MC Dropout configuration affects uncertainty calibration. A grid search shows calibration depends strongly on the interaction between model, dataset, and uncertainty metric, helping explain prior heuristic choices. Nevertheless, across both datasets, applying dropout only to the penultimate layer consistently minimizes monotonicity-violation penalties and yields the most reliable accuracy–uncertainty behavior, rationalizing a common ad-hoc practice (Gu et al., 2024). While this "penultimate-layer rule" is a useful default, it warrants broader validation and a theoretical explanation, including tests on Vision Transformers (Dosovitskiy et al., 2020) and modern CNNs (e.g., EfficientNet (Tan and Le, 2019)), settings beyond classification, calibration when dropout is used during training, and comparisons to alternatives such as multinomial dropout (Li et al., 2019). Practically, improved calibration strengthens selective prediction/deferral workflows (Geifman and El-Yaniv, 2017) by enabling abstention on ambiguous cases and routing them to expert review.

References

Ayer, M., Brunk, H. D., Ewing, G. M., Reid, W. T., and Silverman, E. (1955). An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics*, 26(4):641–647.

Cheddad, A. (2020). Mini-ddsm. Kaggle Datasets. Available at `https://www.kaggle.com/datasets/cheddad/miniddsm2`.

Chernozhukov, V., Fernández-Val, I., and Galichon, A. (2009). Improving point and interval estimates of monotone functions by rearrangement. *Biometrika*, 96(3):559–575.

Corbière, C., Thome, N., Bar-Hen, A., Cord, M., and Pérez, P. (2019). Addressing failure prediction by learning model confidence. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Damianou, A. and Lawrence, N. D. (2013). Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215.

Destounis, S. V., Friedewald, S. M., Grimm, L. J., Poplack, S. P., and Sung, J. S. (2025). Mammography. In *ACR BI-RADS® v2025 Manual*. American College of Radiology, Reston, VA.

DeVries, T. and Taylor, G. W. (2018). Learning confidence for out-of-distribution detection in neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dosovitskiy, A. et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Freeman, I., Roese-Koerner, L., and Kummert, A. (2018). Effnet: An efficient structure for convolutional neural networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 6–10. IEEE.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*.

Gal, Y., Islam, R., and Ghahramani, Z. (2017). Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*.

Geifman, Y. and El-Yaniv, R. (2017). Selective classification for deep neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4885–4894.

Gu, K., Chen, R., and Yao, A. (2024). On the calibration of human pose estimation. In *International Conference on Machine Learning (ICML)*, volume 235, pages 12345–12360.

Guerroudji, M. A. et al. (2014). Segmentation of micro-calcifications in mammograms using mathematical morphology and Otsu's method. *International Journal of Computer Applications*, 975:8887.

Hendrycks, D. and Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Jégou, N. (2012). *Régression isotonique itérée*. PhD thesis, Université Rennes 2.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, ON, Canada.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Li, Y., Gong, X., and Yang, J. (2019). Improved dropout for shallow and deep learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Maleki Sadr, M. A., Gante, J., and Champagne, B. (2022). Uncertainty estimation via monte carlo dropout in cnn-based mmwave mimo localization. *IEEE Wireless Communications Letters*, 11(11):2315–2319.

Medina, M. and co authors (2022). Uncertainty estimation in medical image classification: Systematic review. *JMIR Medical Informatics*, 10(8):e36427.

Nair, T., Precup, D., Arnold, D. L., and Arbel, T. (2020). Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

Valiuddin, H. M., van Sloun, R. J. G., Viviers, C. G. A., de With, P. H. N., and van der Sommen, F. (2024). A review of bayesian uncertainty quantification in deep probabilistic image segmentation. *arXiv preprint arXiv:2411.16370*.

van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. (2020). Uncertainty estimation using a single deep deterministic neural network. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Villanueva Galapon Jr, A., Thummerer, A., Langendijk, J. A., Wagenaar, D., and Both, S. (2024). Feasibility of monte carlo dropout-based uncertainty maps to evaluate deep learning-based synthetic cts for adaptive proton therapy. *Medical Physics*, 51(4):2499–2509.

Wang, J., Ai, J., Lu, M., Liu, J., and Wu, Z. (2023). Predicting neural network confidence using high-level feature distance. *Information and Software Technology*, 159:107214.

Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In *Graphics Gems IV*, pages 474–485. Academic Press Professional, Inc.

Supplementary Materials

A Additional Details on MC Dropout Theory

A.1 Uncertainty Metrics for Classification: Explicit Formulas

This section provides the explicit formulas for the uncertainty metrics introduced in Section 2. Given an input image x, these metrics are computed from p_x, the matrix in $\mathbb{R}^{C\times T}$ whose entry $p_x(i,j)$ is the softmax probability of class i at the j-th stochastic forward pass for input x. Additionally, we introduce p_x^* the vector of length C containing the softmax probabilities of the initial reference pass (i.e., without stochasticity/dropout).

Variation of the predicted class (`variation_predicted`): for a given input image x, measures the dispersion of probabilities of the class predicted at the initial reference pass, noted c_x^*:

$$\mathrm{Var}_{\mathrm{MC}}^{\mathrm{pred}}(p_x, c_x^*) = \frac{1}{T}\sum_{t=1}^{T}(p_x(c_x^*, t) - p_x^*(c_x^*))^2. \tag{9}$$

Variation of the maximum probability (`variation_max`): reflects the stability of the model's highest confidence level across all passes:

$$\mathrm{Var}_{\mathrm{MC}}^{\max}(p_x) = \frac{1}{T}\sum_{t=1}^{T}(\max_c p_x(c,t) - \overline{\max p_x})^2 \tag{10}$$

Where $\max p_x$ is the vector of length T containing the maximum softmax probability for each of the T forward passes, and $\overline{\max p_x}$ is their average.

Predictive entropy (`predictive_entropy`): quantifies the total uncertainty using Shannon entropy (Shannon, 1948) on the averaged predictive distribution:

$$H(p_x) = -\sum_c \overline{p_x(c,.)} \log \overline{p_x(c,.)} \tag{11}$$

Where $p_x(c,.)$ is the c-th line of p_x.

Expected entropy (`expected_entropy`): isolates the data inherent noise (aleatoric uncertainty) by averaging the entropy of each individual stochastic prediction:

$$\overline{H}(p_x) = \frac{1}{T}\sum_{t=1}^{T}\left[-\sum_c p_x(c,t)\log p_x(c,t)\right] \tag{12}$$

BALD (Bayesian Active Learning by Disagreement): isolates the model related noise (epistemic uncertainty) by calculating the mutual information between the predictions and the model parameters (Houlsby et al., 2011; Gal et al., 2017):

$$\mathrm{BALD} = H(p_x) - \overline{H}(p_x) \tag{13}$$

A.2 Calibration Quality Evaluation: Technical Details

This section details the technical foundations and implementation details used to derive the monotonic reference curves and their associated penalties.

A.2.1 Quantile-based Normalization

Directly using the raw uncertainty penalties from Equation 8 makes cross-metric comparisons difficult because of scaling effects. Therefore, we compute the area between the raw and monotonic-correction curves after normalizing the x-axis into dataset quantiles, ensuring that it always represents the same proportion of the dataset. Accuracy computation is as follows. For a threshold τ_q corresponding to the q-th quantile of an uncertainty metric U, we define:

$$\mathcal{S}_q = \{x \in \text{dataset} \mid U(x) \leq \tau_q\} \tag{14}$$

where $q \in \{0.01, 0.02, \ldots, 1.00\}$. This ensures that Accuracy(q) from Equation 7 is evaluated on increasingly large nested subsets $\mathcal{S}_q$ of the data common to all metrics. When $q = 1$, $\mathcal{S}_q$ is the whole dataset.

A.2.2 Visual Comparison of Calibration Quality

Figure 1 illustrates the calibration curves for the CIFAR-10 dataset, a 10-class classification task using the 5-layer CNN architecture detailed in Section B.1. In these examples, uncertainty is quantified via expected entropy.

Figure 1a illustrates a poorly calibrated scenario where dropout is applied across multiple layers with heterogeneous probabilities ($p = 0.7$ for `conv2`, $p = 0.8$ for `conv3`, and $p = 0.4$ for `fc1`). The failure in calibration is evident: rather than following a strictly decreasing trend, the accuracy curve increases across several quantile ranges.

In contrast, Figure 1b shows a well-calibrated curve obtained by restricting dropout to the `fc1` layer (penultimate one) with a probability of $p = 0.8$. This targeted approach yields a nearly monotonic decreasing trend.

A.2.3 Isotonic Regression via the PAV Algorithm (PAVA)

The antitonic (decreasing isotonic) regression is defined as the solution to a constrained least-squares problem. Given the raw accuracy values $a = [a_1, \ldots, a_N]$, we seek $\hat{a}$ such that:

$$\hat{a} = \arg \min_{z_1 \geq z_2 \geq \cdots \geq z_N} \sum_{k=1}^{N} (a_k - z_k)^2 \tag{15}$$

We compute this using the *Pool Adjacent Violators Algorithm (PAVA)* (Ayer et al., 1955; Jégou, 2012), following these steps:

1. **Initialize**: Start with the raw sequence of accuracies.
2. **Find Violations**: Identify any pair (a_k, a_{k+1}) where $a_k < a_{k+1}$ (violating the decreasing constraint).
3. **Pool**: Replace the violators with their weighted average.
4. **Iterate**: Repeat until the entire sequence is monotonically non-increasing.

This method acts as a non-parametric regularizer, rectifying any local increases in accuracy caused by sampling noise (see Figure 4).

A.2.4 Monotonic Rearrangement

In contrast to Isotonic Regression, which modifies the values themselves, the monotonic rearrangement (Chernozhukov et al., 2009) preserves the original values of the curve but reassigns them to the quantiles in a non-increasing order. Let $\mathcal{A} = \{a_1, \ldots, a_N\}$ be the set of observed accuracies. The rearranged curve $a^*(x)$ is obtained by sorting $\mathcal{A}$ such that:

$$a_1^* \geq a_2^* \geq \cdots \geq a_N^* \tag{16}$$

A key property of this approach is that it is guaranteed to be closer to the underlying monotonic function under any L_p norm (Chernozhukov et al., 2009).

The key distinctions between the two correction methods are summarized below (see Table 2):

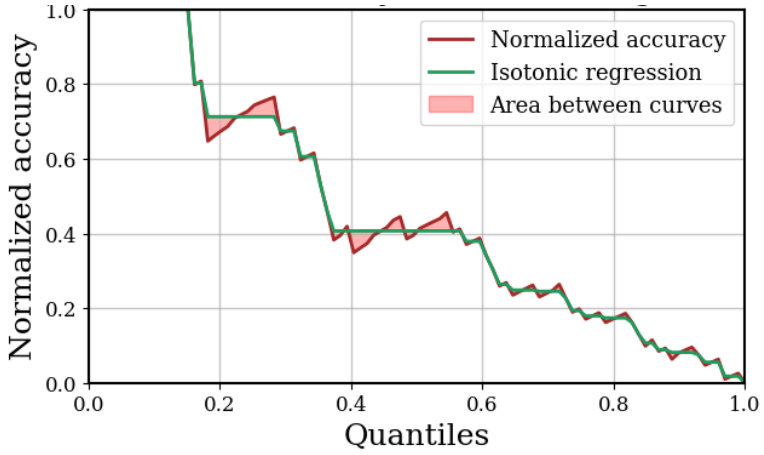


Figure 4: Isotonic correction and monotonicity penalty (red area) based on `expected_entropy` (see Supplementary A.1) for a well-calibrated dropout configuration. The gaps illustrate the non-monotonic behavior. Configuration: dropout at `fc1` ($p = 0.8$) within a CNN (3 conv, 2 fc). Dataset: CIFAR-10.

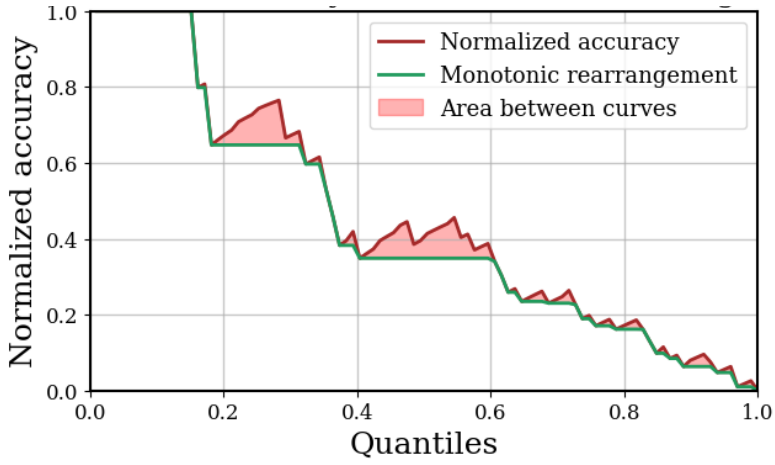


Figure 5: Monotonic rearrangement and monotonicity penalty (red area) based on `expected_entropy` (see Supplementary A.1) for a well-calibrated dropout configuration. The gaps illustrate the non-monotonic behavior. Configuration: dropout at `fc1` ($p = 0.8$) within a CNN (3 conv, 2 fc). Dataset: CIFAR-10.

Table 2: Comparison of Monotonic Correction Methods Used for Penalty Calculation

Feature	**Isotonic Regression**	**Monotonic Rearrangement**
Optimization	Minimizes L_2 distance to raw data	Minimizes L_p distance to target function
Values	Values are averaged (modified)	Values are sorted (preserved)
Effect	Smooths the curve	Reorders the curve

A.2.5 Approximation of the Area between the Curves

The monotonicity penalty is calculated as the discrete approximation of the integral over the $N = 100$ quantiles:

$$\text{Penalty} \approx \frac{1}{N} \sum_{k=1}^{N} |a_k - c_k| \tag{17}$$

The monotonicity penalties quantitatively confirm the sharp disparity in calibration quality between the two configurations of Figure 1 :

Table 3: Comparison of Monotonicity Penalties for chosen Dropout Configurations represented in Figure 1. Dataset: CIFAR-10

Dropout Configuration	Isotone Penalty	Rearrangement Penalty
fc1: 0.8	0.002668	0.005607
conv2: 0.7, conv3: 0.8, fc1: 0.4	0.030307	0.469543

B Implementation Details

All experiments were conducted using a Python 3.13 kernel and Visual Studio Code IDE on a machine equipped with an NVIDIA RTX 4060 GPU. Code is available at `https://anonymous.4open.science/r/MonteCarlo_Dropout`.

B.1 Datasets and Models

This subsection provides the exhaustive architectural and training specifications for the models introduced in the main text.

B.1.1 CIFAR-10: 5-Layer CNN Specifications

The custom architecture used for the initial discovery phase is a sequential CNN designed for 32×32 RGB inputs. The precise layer configuration is as follows:

- **Conv Block 1:** 16 filters (3×3), padding 1, ReLU, 2×2 max-pooling.
- **Conv Block 2:** 32 filters (3×3), padding 1, ReLU, 2×2 max-pooling.
- **Conv Block 3:** 64 filters (3×3), padding 1, ReLU, 2×2 max-pooling.
- **Fully-Connected 1:** $1024 \to 128$ units, ReLU.
- **Output Layer:** $128 \to 10$.

The model is trained on the CIFAR-10 dataset partitioned into 45,000 images for training, 5,000 for validation, and 10,000 for testing. Input images are normalized with a mean and standard deviation of 0.5 for each RGB channel. Optimization is performed using the Adam optimizer with a learning rate of $\eta = 0.001$ and a standard BCE loss function, using a batch size of 128. To ensure optimal generalization, the final model parameters are selected based on the highest validation accuracy attained during the training process. Under these settings, and with the test set balanced by default, the baseline test accuracy (without dropout) was 72.67%.

B.1.2 Mini-DDSM: EffNet for Grayscale Mammography

For the abnormality detection task (Normal vs. Abnormal), we adapted the EffNet architecture (Freeman et al., 2018) to handle single-channel grayscale inputs (224×224). The binarization process grouped "Benign" and "Cancer" cases into the "Abnormal" class (0) vs the "Normal" class (1).

The specialized blocks consist of:

- **Stem:** $1 \to 32$ channels, 3×3 kernel, LeakyReLU, Batch Normalization.
- **EffNet Blocks ($\times 3$):** Successive 1×1 expansions, 1×3 and 3×1 depthwise convolutions, and asymmetrical pooling to capture features across different scales. Channels expand as $32 \to 64 \to 128 \to 256$.
- **Classifier:** Global average pooling followed by a linear layer to 2 units.

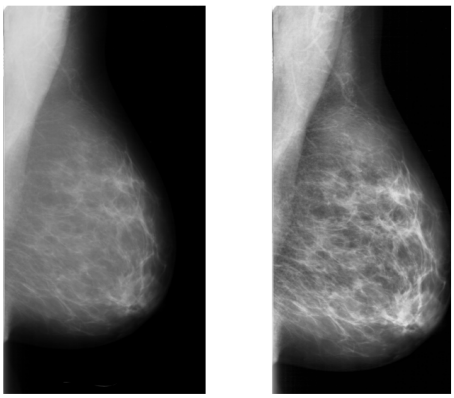
The dataset was partitioned into training ($N = 6326$), validation ($N = 703$), and test ($N = 781$) sets. The model was trained for 20 epochs using the Adam optimizer with an initial learning rate $\eta = 10^{-3}$ and a cosine-annealing scheduler to adjust the learning rate over time. We used the BCE loss and Automated Mixed Precision (AMP) for efficiency, and employed a sampler to mitigate class imbalance during training by undersampling the majority class (abnormal mammograms, labeled 0). Final performance was evaluated on a balanced test set (50% per class) to ensure that the 87.20% baseline accuracy reflects diagnostic performance rather than class prevalence.

B.2 Mammograms Preprocessing

Inspired by segmentation techniques for micro-calcifications (Guerroudji et al., 2014), our preprocessing pipeline begins by converting images to grayscale and cropping a 30-pixel border to remove common scanning artifacts. We employ Otsu thresholding (Otsu, 1979) to generate a foreground mask, followed by three iterations of erosion and five iterations of dilation to eliminate isolated noise. Saturated components, defined as regions where over 90% of pixels exceed a value of 245, are systematically removed. To isolate the breast tissue, we use row and column projections with a threshold set at 3% of the maximum sum, adding an 8% buffer to the resulting bounding box to ensure no peripheral tissue is lost.

The localized tissue then undergoes CLAHE (Contrast Limited Adaptive Histogram Equalization) (Zuiderveld, 1994) with a clip limit of 2.0 and 8×8 tiles to enhance local contrast, followed by z-score normalization. Finally, all images are resized to 200×200 pixels. Figure 6 illustrates these transformations on a medio-lateral oblique view of a right breast mammogram.

For the training phase, we implement a data augmentation strategy to limit overfitting on the minority class (normal mammograms). This includes random rotations within $\pm 15°$, random resized crops (scale 0.9–1.0), and a 50% horizontal flip. We also apply brightness and contrast jitter (0.1), alongside conditional Gaussian blurring (using kernels of size 5 or 3 with $p = 0.3$) and sharpness adjustments (factor 1.5, $p = 0.3$). In contrast, the test and validation phases involve only resizing and normalization, using the mean and standard deviation computed across the entire training dataset.



(a) Mammogram Before Preprocessing (b) Mammogram After Preprocessing

Figure 6: Preprocessing visualization of one randomly picked mammogram. Laterality: right, view: medio-lateral oblique (MLO). Preprocessing visibly enhances contrast between fatty and fibrograndular tissues. Dataset: Mini-DDSM.

C Additional Experimental Results

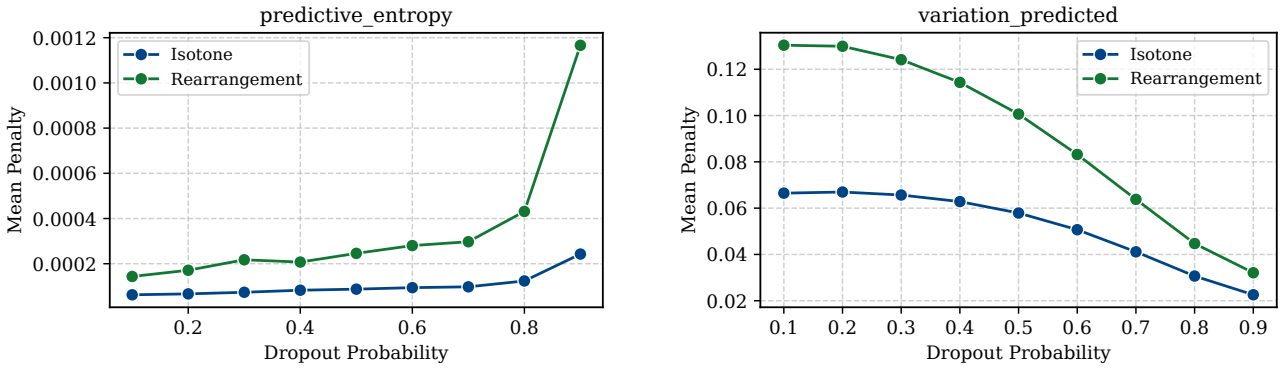
C.1 Impact of Dropout Probability on Calibration

To evaluate the impact of dropout probabilities across different architectural depths, we define the mean dropout probability, p_{mean}, for a given configuration as the average of the individual dropout probabilities applied across

the modified layers:

$$p_{\text{mean}} = \frac{1}{L} \sum_{h=1}^{L} p_h \tag{18}$$

where L denotes the total number of layers subject to dropout in the specific configuration, and p_h represents the dropout probability applied to the h-th layer.



(a) Degradation of predictive entropy.

(b) Improvement of variation predicted.

Figure 7: Contrasting effects of mean dropout probability (p_{mean}) on calibration metrics, evaluated through isotone regression and monotonic rearrangement penalties. Figure 7a highlights that for the `expected_entropy` metric, higher dropout probability increases monotonicity penalties, signaling calibration degradation. Conversely, Figure 7b shows that `variation_predicted` is better calibrated (lower monotonicity violation penalties) when the mean dropout across masked neurons is higher. Configuration: CNN (3 conv, 2 fc). Dataset: CIFAR-10.

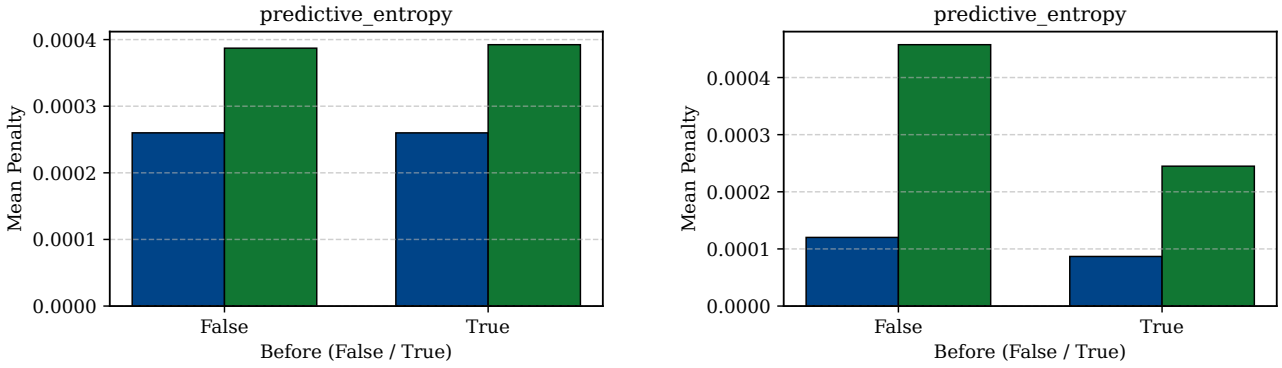
C.2 Impact of Placement and Number of Layers on calibration

C.2.1 Dropout Placement

To analyze the impact of dropout placement, we defined a boolean flag, `before`, indicating whether the dropout mask is applied before (`True`) or after (`False`) a given activation layer. The results were grouped by the average dropout probability across all layers. The monotonicity penalties were aggregated conditionally based on this placement:

$$\text{penalty}(b) = \mathbb{E}\big[\text{penalty} \mid \texttt{before} = b\big]$$

where $b \in \{\text{True}, \text{False}\}$ denotes the dropout position.



(a) Mini-DDSM dataset.

(b) CIFAR-10 dataset.

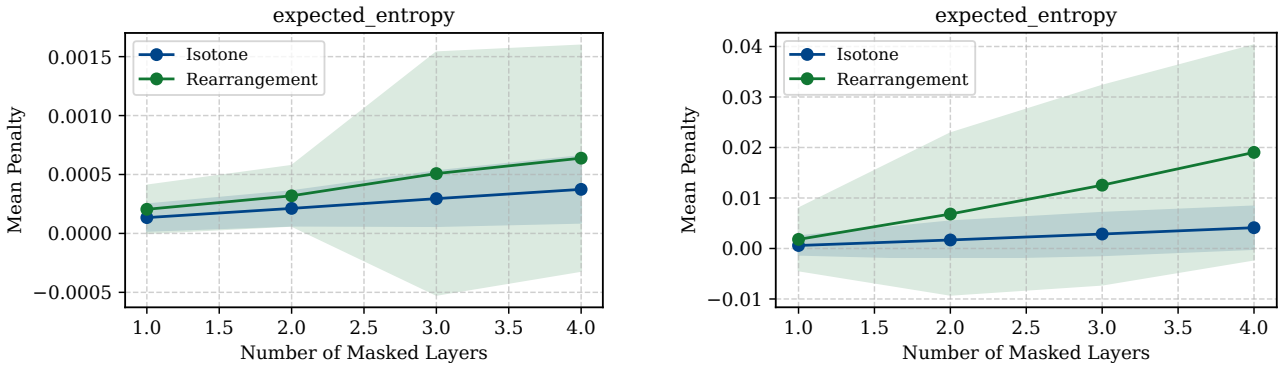
Figure 8: Impact of dropout placement (Before vs. After activation) on predictive entropy calibration across Mini-DDSM (8a) and CIFAR-10 (8b) datasets. Results are aggregated by the mean dropout probability, where the boolean flag `before` indicates whether the dropout mask is applied before (`True`) or after (`False`) the activation layer. The plots show the expected monotonicity penalties for each position. While some trends appear, the effect of placement varies across different uncertainty metrics and the observed differences are not statistically significant.

C.2.2 Number of Masked Layers

To evaluate the effect of the number of active dropout layers, we computed the total number of layers with a non-zero dropout probability for each configuration:

$$n_{\text{layers}} = \#\{\, \text{layer } l \mid p_l > 0 \,\}$$

The monotonicity penalties (isotone and rearrangement) were extracted for each metric. Finally, the results were grouped by n_{layers} to calculate the mean and standard deviation of the penalties across all grid search runs.



(a) Mini-DDSM dataset.

(b) CIFAR-10 dataset.

Figure 9: Impact of the number of active dropout layers (n_{layers}) on expected entropy calibration. For each configuration, n_{layers} is defined as the count of layers with a non-zero dropout probability. The plots show the mean monotonicity penalties (isotone regression and monotonic rearrangement) averaged across the evaluated model configurations. However, the observed trends are not statistically significant, as the high standard deviation (shaded areas) indicates a lack of consistent impact across different configurations and uncertainty metrics.

C.3 Statistical Analysis of Probabilities Distribution

To better understand the structural impact of dropout across different depths of the network, we quantify the asymmetry and concentration of the dropout probabilities across the layers. We consider the layers sequentially, assigning an index $l \in \{1, 2, 3, 4\}$ corresponding to `conv1`, `conv2`, `conv3`, and `fc1`, respectively.

C.3.1 Probability Normalization.

For each configuration, we first construct a normalized probability distribution p'_l over the network layers:

$$p'_l = \frac{p_l}{\sum_j p_j} \tag{19}$$

where p_l is the original dropout probability applied to layer l. Based on this normalized distribution, we compute the weighted mean μ and variance σ^2 of the layer indices as:

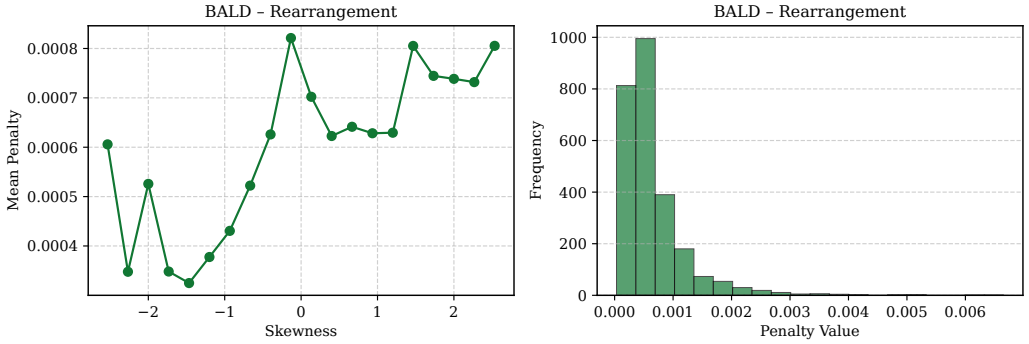
$$\mu = \sum_l p'_l \, l \quad \text{and} \quad \sigma^2 = \sum_l p'_l \, (l - \mu)^2 \tag{20}$$

C.3.2 Skewness

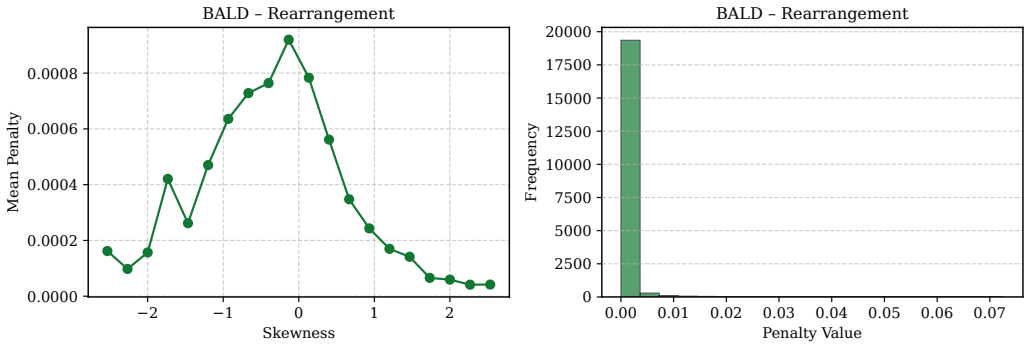
To quantify the asymmetry of the dropout distribution along the network depth, we calculate the weighted skewness, defined as the third standardized moment of the layer indices:

$$\text{Skewness} = \sum_l p'_l \left(\frac{l - \mu}{\sigma} \right)^3 \tag{21}$$

This metric allows us to observe whether the dropout probabilities are skewed toward the earlier or deeper layers of the architecture. Across architectures, skewness showed only inconsistent, configuration-dependent correlations with the monotonicity penalty, and did not generalize as a reliable indicator across both CIFAR-10 and Mini-DDSM.



(a) Mini-DDSM dataset.



(b) CIFAR-10 dataset.

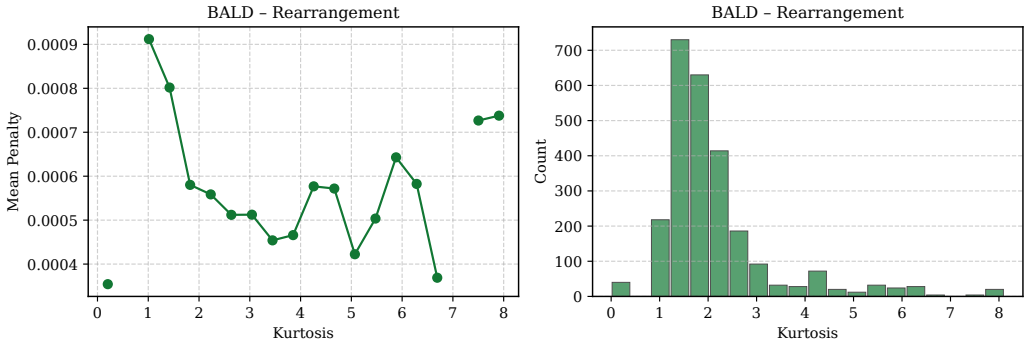
Figure 10: Impact of dropout probability skewness on BALD rearrangement penalty. Skewness is computed as the third standardized moment of the normalized layer indices p'_l. Figure 10a highlights a high degree of variability and noise for the Mini-DDSM dataset, while Figure 10b shows a clear peak in penalty for symmetric distributions (skewness ≈ 0), with the penalty decreasing as the distribution becomes asymmetric. The plots illustrate the mean penalty across various configurations. However, no statistically significant trend is observed globally, as the impact of skewness remains inconsistent across different uncertainty metrics and exhibits high variability between datasets.

C.3.3 Kurtosis

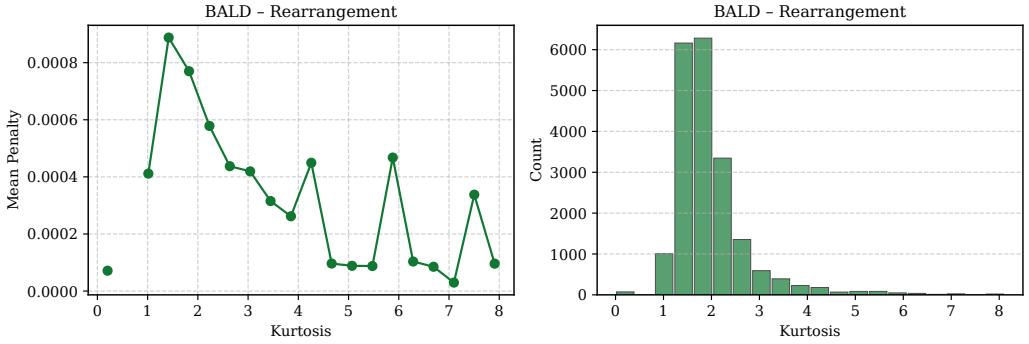
Similarly, to measure the concentration or dispersion of the dropout probabilities, we compute the weighted kurtosis (the fourth standardized moment):

$$\text{Kurtosis} = \sum_l p'_l \left(\frac{l - \mu}{\sigma} \right)^4 \tag{22}$$

A high kurtosis indicates a highly concentrated probability distribution (e.g., sharp peaks at specific layers), whereas a low kurtosis reflects a flatter, more uniform distribution of dropout probabilities across the network. As with skewness, kurtosis exhibited only localized, architecture-specific trends and its correlation with monotonicity penalty was inconsistent, providing no universal performance indicator on either CIFAR-10 or Mini-DDSM.



(a) Mini-DDSM dataset.



(b) CIFAR-10 dataset.

Figure 11: Impact of dropout probability kurtosis on BALD rearrangement penalty. Kurtosis is computed as the fourth standardized moment of the normalized layer indices p'_l. For both Figures 10a and 10b, no statistically significant results are observed, as the penalty values exhibit high variability and lack a consistent trend across the evaluated kurtosis range.

C.4 Dropout Depth

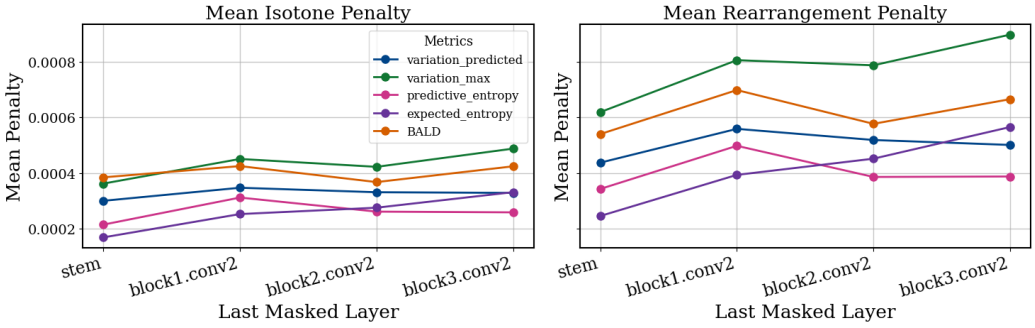
To investigate the sensitivity of calibration penalties to the depth at which dropout is introduced or terminated, we define specific indicators based on the position of active dropout layers within the predefined hierarchies.

For a given dropout configuration $d \subset \mathcal{L}$, where $\mathcal{L}$ is the set of candidate layers, we identify the first and last layers subjected to dropout according to their sequential index:

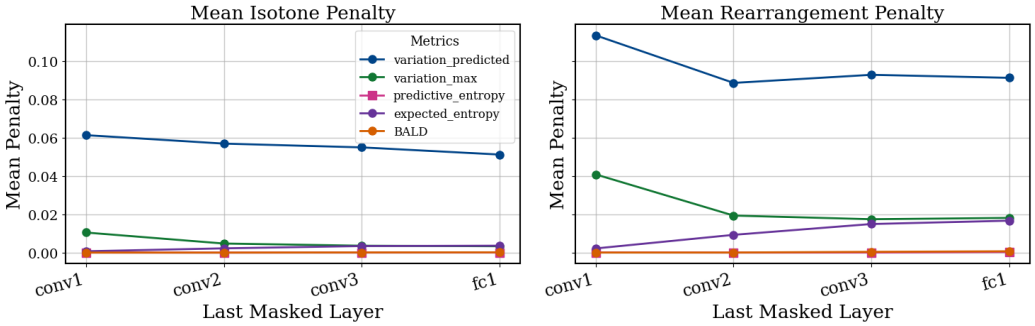
$$l_{\text{first}} = \arg\min_{l \in d} \text{index}(l) \quad \text{and} \quad l_{\text{last}} = \arg\max_{l \in d} \text{index}(l) \tag{23}$$

To quantify the impact of architectural depth on calibration, we aggregate monotonicity penalties (isotone and rearrangement) by their entry (l_{first}) and exit (l_{last}) layers. For each unique (layer, probability) pair, we compute

the mean and standard deviation (mean $\pm$ std), reducing the high-dimensional configuration space into marginal distributions. This allows for a direct comparison of penalty sensitivity between early-stage (e.g., `stem`) and late-stage (e.g., `block3`) dropout regularization.



(a) Mini-DDSM dataset.



(b) CIFAR-10 dataset.

Figure 12: Impact of the last masked layer (l_{last}) on calibration penalties. Here, l_{last} represents the deepest layer in the network with a non-zero dropout probability. Figure 12a highlights that for DDSM, penalties increase as l_{last} moves deeper into the architecture, whereas Figure 12b shows metric-dependent sensitivity for CIFAR-10. For both figures, no statistically significant results are observed.