ATTENUATE, DON'T DROP: A SIGN ENTROPY APPROACH TO WEIGHT REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern neural networks often achieve high accuracy while producing poorly calibrated confidence estimates. Regularization techniques like Dropout and DropConnect improve generalization by randomly masking activations or weights. However, they do not address calibration error. To address this, we propose a stability-guided posterior attenuation mechanism for Bayesian Neural Networks (BNNs). We use Sign Entropy, an information-theoretic quantity, to identify unstable weights and perform annealed attenuation on them. Sign Entropy measures the posterior uncertainty of the sign of each Bayesian weight, i.e., whether the weight provides positive or negative evidence. During training, we softly attenuate high Sign Entropy weights by jointly scaling their posterior mean and standard deviation. We provide theoretical justification by showing that, in the single-feature binary classification case, Sign Entropy is directly related to the posterior entropy of the predicted class, and we extend it to multi-feature settings. We demonstrate our results on classification and regression tasks, using two well-known Deep Learning (DL) backbones on two datasets, and show that the proposed method generally preserves predictive accuracy while improving probabilistic fit, calibration error, epistemic uncertainty, and predictive interval quality.

1 INTRODUCTION

Modern deep neural networks often achieve high predictive accuracy but can still produce poorly calibrated confidence estimates, where predicted confidence does not match empirical correctness (Guo et al., 2017; Gal & Ghahramani, 2016). This issue is particularly important in settings where predictions are used together with uncertainty estimates. Regularization methods such as Dropout (Srivastava et al., 2014) and DropConnect (Wan et al., 2013) are widely used to improve generalization by injecting stochasticity during training but do not address the issue of poor calibration. Dropout randomly suppresses activations, while DropConnect randomly suppresses weights. Concrete Dropout (Gal et al., 2017) extends this idea by learning the dropout probability through a continuous relaxation of the Bernoulli mask, making the regularization strength adaptive rather than fixed. Although these mechanisms are effective, they are largely uninformed i.e., a random mask is applied across all activations or weights across different epochs during training.

Bayesian neural networks (BNNs) (Blundell et al., 2015; Gal & Ghahramani, 2016) provide a natural way to represent epistemic uncertainty by maintaining posterior distributions over activations and weights. However, in many BNN settings, the posterior uncertainty is mainly used for estimating prediction uncertainty after training. Less attention has been given to using posterior uncertainty itself as a signal for guiding regularization during training. This motivates the central question of this work: *can posterior information about the directional stability of weights be used to regularize a model deterministically than random masking?*

To answer this question we propose a Sign Entropy guided regularization which performs train-time attenuation of weights for BNNs. For each Bayesian weight, we compute the posterior probability that the weight is positive or negative (i.e., contribution/influence) and compute its Sign Entropy as the uncertainty of the posterior sign of the weight. A high Sign Entropy indicates that the model is uncertain about whether the corresponding weight provides positive or negative contribution. Instead of randomly dropping weights in a stochastic, our method attenuates weights with high posterior sign uncertainty during training in a deterministic manner. Thus, the proposed method is both deterministic

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

and uncertainty-guided, in contrast to the random masking used by other approaches (e.g., Dropout, DropConnect, and Concrete Dropout).

The deterministic and uncertainty guided nature of the proposed method hinges on Sign Entropy as a measure to identify unstable weights. It then attenuates both the posterior mean and posterior standard deviation of the identified weights. This reduces the influence of directionally unstable weights on the model output. We theoretically motivate this mechanism by showing that in a single-feature binary classification case, Sign Entropy is directly linked to the uncertainty of the predicted class. We then extend it to multiple feature settings, where posterior uncertainty in Bayesian layer weights induces logit variation. This logit variation can propagate through the activation function contributing to uncertainty in prediction.

Our work contributes to:

- A novel Sign Entropy guided train-time attenuation mechanism for BNNs based on theoretical justification linking posterior sign uncertainty to predictive class instability through a single-feature analysis and a multi-feature logit-variance extension.
- Establishing empirical evidence for the effectiveness of the proposed approach on both classification and regression tasks. This is validated on two well-known backbones and across two datasets using multiple measures.

For classification, we study predictive performance and calibration using macro-F1, negative log-likelihood (NLL), Brier score, Adaptive Expected Calibration Error (AECE), and epistemic uncertainty. For regression, we evaluate predictive accuracy and uncertainty using Mean Absolute Error (MAE), NLL, Prediction Interval Coverage Probability (PICP) and Mean Prediction Interval Width (MPIW). Across these settings, the proposed method generally preserves predictive accuracy while improving probabilistic metrics such as NLL, Brier score, calibration error, and epistemic uncertainty compared with standard train-time regularizers. We provide ablation studies for hyperparameter sensitivity and the attenuation mechanism.

2 Theoretical Background

Preliminaries. Let x denote an input sample and let $h(x) \in \mathbb{R}^d$ denote the feature vector produced before the Bayesian layer on which we want to perform the attenuation. We denote $h_j(x)$ as the activation for the j-th feature. Each weight has a Gaussian variational posterior: $w_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$.

For binary classification, the final logit can be written as $z(x) = \sum_{j=1}^{d} w_j h_j(x) + b$, where $z(x)$ is the scalar logit, $h_j(x)$ is the j-th feature activation, w_j is the corresponding final-layer Bayesian weight, and b is the bias term. The predicted probability therefore becomes:

$$\hat{p}(x) = \mathrm{sig}(z(x)) = \frac{1}{1 + \exp(-z(x))}, \tag{1}$$

where $\mathrm{sig}(\cdot)$ is the sigmoid function[1].

The predicted class $\hat{Y}$ is given as:

$$\hat{Y} = \mathbf{1}\{z(x) \geq 0\}, \tag{2}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. Thus, the sign of the logit determines the predicted class.

Sign Entropy (Bora et al., 2025b). For each Bayesian weight $w_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$, the posterior probability that the weight is positive or negative is computed as:

$$p_j^+ = \mathbb{P}(w_j \geq 0) = \Phi\left(\frac{\mu_j}{\sigma_j}\right), \qquad p_j^- = \mathbb{P}(w_j < 0) = \Phi\left(-\frac{\mu_j}{\sigma_j}\right), \tag{3}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, p_j^+ is the probability that weight j is positive, and p_j^- is the probability that weight j is negative.

[1]We are using Sigmoid as an illustrative case. However, it can be extended to other activations.

Sign Entropy of weight j is defined as

$$H_s(w_j) = -p_j^+ \log_2 p_j^+ - p_j^- \log_2 p_j^- . \tag{4}$$

Here, $H_s(w_j)$ measures uncertainty in the *direction* of the feature effect. If $H_s(w_j)$ is close to zero, the posterior assigns almost all probability to one sign, i.e., the feature has a stable directional contribution. If $H_s(w_j)$ is close to one, then $p_j^+ \approx p_j^- \approx 0.5$, i.e., the model is uncertain whether the corresponding feature contributes positively or negatively.

Single-feature case: Sign Entropy equals predictive class entropy. Consider the simplest case with one feature, no bias, and a positive activation ($h(x) > 0$). The logit is then

$$z(x) = wh(x). \tag{5}$$

Since $h(x) > 0$, the sign of the logit is directly linked to the sign of the weight[2]:

$$\text{sign}(z(x)) = \text{sign}(w). \tag{6}$$

Therefore using Equation (2) we have,

$$\hat{Y} = \mathbf{1}\{z(x) \geq 0\} = \mathbf{1}\{w \geq 0\}. \tag{7}$$

Hence, under the posterior over w, the predicted class is a Bernoulli random variable:

$$\mathbb{P}(\hat{Y} = 1 \mid x) = p^+, \qquad \mathbb{P}(\hat{Y} = 0 \mid x) = p^-. \tag{8}$$

The entropy or uncertainty of the predicted class is therefore

$$H(\hat{Y} \mid x) = -p^+ \log_2 p^+ - p^- \log_2 p^- = H_s(w). \tag{9}$$

Thus from Equation (9) we can see that, in the single-feature case, Sign Entropy is the posterior entropy of the predicted class.

Multi-feature extension. For multiple features, Equation (9) can be extended (details in Section A1.1 of Appendix) to define the entropy of the predicted class as:

$$H(\hat{Y} \mid x) = H_s\left(\Phi\left(\frac{|m(x)|}{s(x)}\right)\right), \tag{10}$$

Where, $m(x)$ is the posterior mean of the logit and $s(x)$ is the posterior standard deviation of the logit.

High uncertainty arises when the posterior logit distribution places substantial mass on both sides of the decision boundary. The per-weight Sign Entropy $H_s(w_j)$ is an input-independent proxy for this logit-level uncertainty. A high Sign Entropy weight is directionally unstable: under posterior sampling it can switch between positive and negative contributions. If many such weights contribute to the final logit, the sign of the logit can become unstable, especially near the decision boundary.

From logit variation to confidence variation. The predicted probability or confidence is determined by the sigmoid of the logit. The first order derivative of $\text{sig}(z)$, $\text{sig}'(z)$

$$\text{sig}'(z) = \text{sig}(z)\,(1 - \text{sig}(z)) \leq \frac{1}{4}, \tag{11}$$

we have the global Lipschitz bound $|\text{sig}(z) - \text{sig}(z')| \leq \frac{1}{4}|z - z'|$. This means that variation in the logit induces variation in predicted confidence. The effect is strongest near the decision boundary, where $z \approx 0$ and the slope of the sigmoid is the largest.

For small logit variance, a local delta-method approximation (Oehlert, 1992) gives

$$\text{Var}(\hat{p} \mid x) \approx [\text{sig}(m(x))\,(1 - \text{sig}(m(x)))]^2\, s^2(x). \tag{12}$$

[2]If $h(x) < 0$, the sign relation is reversed, so $\mathbb{P}(\hat{Y} = 1 \mid x) = \mathbb{P}(w < 0) = p^-$ instead of p^+; however, since $p^- = 1 - p^+$ and binary entropy is symmetric, the predictive class entropy remains equal to the Sign Entropy.

Here, $\mathrm{Var}(\hat{p} \mid x)$ is the posterior variance of the predicted probability, $m(x)$ is the posterior mean logit, and $s^2(x)$ is the posterior logit variance. This approximation shows that logit variance is most relevant for confidence variation when the sample lies near the decision boundary.

When prediction is performed by Monte Carlo marginalization, posterior class instability directly affects the reported predictive distribution. If high sign entropy weights cause different posterior samples to favor different classes, the averaged predictive probability becomes less stable and the confidence assigned to the final prediction changes. Since calibration compares predicted confidence with empirical correctness, attenuating such unstable posterior class fluctuations provides a direct mechanism by which sign entropy guided attenuation can improve calibration. This mechanism links sign entropy to calibration relevant predictive uncertainty, which can be quantified using NLL, Brier score, and adaptive calibration error.

3 Proposed Approach

Overview. Section 2 provides background for using Sign Entropy as a measure of directional uncertainty in Bayesian layers. We now describe how this quantity is used during training. The proposed method is a train-time attenuation mechanism: instead of randomly dropping activations as in Dropout or randomly dropping weights as in DropConnect, we attenuate final-layer Bayesian weights whose posterior sign is uncertain.

For notational simplicity, let j denote an index of the flattened final-layer weight. In a multi-class classifier, j corresponds to a pair (k, l), where k is the output class and l is the input feature index. Each Bayesian weight has posterior parameters (μ_j, σ_j), where μ_j is the posterior mean and σ_j is the posterior standard deviation. We now describe the main stages in our proposed approach as below:

3.1 Sign Entropy guided selection.

Let $\rho \in (0, 1)$ be the attenuation ratio (equivalent to dropout ratio in classical Dropout or DropConnect). For each final-layer Bayesian weight w_j, we compute its Sign Entropy score

$$E_j = H_s(w_j), \tag{13}$$

where E_j measures the posterior uncertainty about the sign of weight j. A larger E_j indicates that the model is more uncertain whether the weight provides positive or negative contribution.

We flatten the Sign Entropy scores of the Bayesian layer and compute the quantile threshold

$$\tau = Q_{1-\rho}\left(\{E_j\}_{j=1}^{d}\right), \tag{14}$$

where $Q_{1-\rho}$ denotes the $(1-\rho)$ quantile and d is the number of weights in the flattened Bayesian layer. We then define the binary selection mask

$$M_j = \mathbf{1}\{E_j > \tau\}. \tag{15}$$

Here, $M_j = 1$ means that weight j belongs to the high Sign Entropy group selected for attenuation, while $M_j = 0$ means that it is not selected.

3.2 Soft attenuation gate.

For the selected weights 'M_j' we compute a soft gate

$$g_j = \mathrm{sigmoid}\left(\alpha(E_j - \tau)\right) M_j, \tag{16}$$

where $\alpha > 0$ is the gate sharpness. A larger α makes the transition sharper, while a smaller α makes the transition softer. Since $E_j \in [0, 1]$, the gate operates on a bounded. The multiplication by M_j ensures that only weights above the quantile threshold are attenuated.

3.3 Annealing over training.

To avoid imposing strong attenuation at the beginning of training, the gate is annealed over epochs:

$$\eta(t) = 1 - \left[\frac{1}{2}\left(1 + \cos\left(\frac{\pi t}{T}\right)\right)\right]^{\gamma}, \tag{17}$$

where t is the current epoch, T is the total number of training epochs, and $\gamma > 0$ controls the shape of the schedule. At the beginning of training, $\eta(0) = 0$ i.e., no attenuation is applied and at the end of training, $\eta(T) = 1$, the full attenuation strength is reached.

The final attenuation coefficient for weight j at epoch t is $a_j(t) = \eta(t)g_j$, and the corresponding retain factor is $R_j(t) = 1 - a_j(t)$ for the weight.

3.4 Joint posterior attenuation.

The proposed method jointly attenuates the posterior mean and posterior standard deviation:

$$\widetilde{\mu}_j = R_j(t)\mu_j, \qquad \widetilde{\sigma}_j = R_j(t)\sigma_j. \tag{18}$$

Here, $\widetilde{\mu}_j$ and $\widetilde{\sigma}_j$ are the attenuated posterior mean and posterior standard deviation. Thus, the joint posterior attenuation reduces both the expected contribution of the weight and its posterior spread.

An important consequence of joint attenuation is that the posterior relationship remains unchanged whenever $R_j(t) > 0$ as shown below:

$$\frac{|\widetilde{\mu}_j|}{\widetilde{\sigma}_j} = \frac{|R_j(t)\mu_j|}{R_j(t)\sigma_j} = \frac{|\mu_j|}{\sigma_j}. \tag{19}$$

As it can be noted, the proposed method does not stabilize a directionally unstable weight. Instead, it reduces the influence of that unstable weight on the final prediction. A pictorial representation of the proposed approach is provided in Figure A1 in the Appendix.

4 Experimental Setup

4.1 Datasets and Models

We evaluate the proposed Sign-Entropy-guided posterior attenuation mechanism on both classification and regression tasks. For classification, we use Fashion-MNIST, a grayscale image classification benchmark containing clothing categories (Xiao et al., 2017). For regression, we use the UTKFace dataset for age estimation, where the task is to predict a continuous age value from a face image (Zhang et al., 2017).

For both tasks, we use two convolutional backbone architectures: VGG16 (Simonyan & Zisserman, 2015) and ResNet18 (He et al., 2016). In each case, the deterministic backbone is followed by a Bayesian final layer, following the Bayesian neural network formulation of variational weight uncertainty (Blundell et al., 2015). The proposed approach is applied only to this final Bayesian layer. This choice isolates the effect of posterior attenuation at the decision layer while keeping the feature extractor architecture unchanged. For classification, the final Bayesian layer outputs class logits and for regression, we use a Bayesian split-head formulation for age prediction[3]. For both the tasks, we compare the proposed method against a plain Bayesian neural network, Dropout (Srivastava et al., 2014), DropConnect (Wan et al., 2013), and Concrete Dropout (Gal et al., 2017).

4.2 Evaluation Metrics

For classification, we report macro-F1, negative log-likelihood (NLL), multiclass Brier score, adaptive expected calibration error (AECE), and epistemic uncertainty. Macro-F1 measures class-balanced predictive performance (van Rijsbergen, 1979). NLL and Brier score evaluate the quality of the predicted probability distribution (Bishop, 2006; Brier, 1950), while AECE measures confidence calibration using adaptive confidence bins (Mukhoti et al., 2020). The epistemic uncertainty is estimated from Monte Carlo posterior samples using the standard predictive entropy decomposition (Houlsby et al., 2011; Kendall & Gal, 2017).

For regression, we report Mean Ebsolute Error (MAE), Gaussian NLL, age-bin Brier score, epistemic uncertainty, absolute coverage error at 90% Predictive Interval Coverage (CE90), and Mean Prediction Interval Width at 90% coverage (MPIW90). MAE measures point prediction accuracy, while NLL

[3]Details of implementation can be found in the anonymous link: `https://anonymous.4open.science/r/ICLR2027-B42D/`

Table 1: Fashion-MNIST classification results for ResNet18 and VGG16 with initial/drop ratios $\rho = 0.1$, and $\rho = 0.3$. Higher F1 is better; lower NLL, Brier, AECE, and Epistemic uncertainty (denoted by Epistemic U) are better. The best scores within each architecture/drop ratio block are highlighted in bold.

Method	F1 ↑	NLL ↓	Brier ↓	AECE ↓	Epistemic U ↓
VGG16, initial/drop ratio $\rho = 0.1$					
Concrete Dropout	0.9218	0.3798	0.1466	0.1604	0.000070
Plain BNN	0.9093	0.4262	0.1685	0.1754	0.002011
Dropout	0.9050	0.4375	0.1740	0.1750	0.002298
DropConnect	0.8808	0.5272	0.2240	0.1817	0.059609
Proposed	**0.9247**	**0.3283**	**0.1306**	**0.1235**	**0.000020**
ResNet18, initial/drop ratio $\rho = 0.1$					
Plain BNN	0.9374	0.3359	0.1248	0.1622	0.004080
Dropout	0.9412	0.3845	0.1402	0.2071	0.003607
DropConnect	**0.9420**	0.3776	0.1415	0.1934	0.023596
Proposed	0.9413	**0.2373**	**0.0936**	**0.0566**	**0.000040**
VGG16, initial/drop ratio $\rho = 0.3$					
Concrete Dropout	**0.9222**	0.4040	0.1567	0.1853	0.000101
Plain BNN	0.9081	0.4289	0.1703	0.1783	0.001995
Dropout	0.8987	0.4516	0.1813	0.1759	0.002055
DropConnect	0.8891	0.4908	0.2064	0.1753	0.036521
Proposed	0.9217	**0.3764**	**0.1446**	**0.1578**	**0.000023**
ResNet18, initial/drop ratio $\rho = 0.3$					
Concrete Dropout	**0.9488**	0.3149	0.1091	0.1661	0.000399
Plain BNN	0.9370	0.3375	0.1249	0.1598	0.003516
Dropout	0.9327	0.5063	0.1930	0.2793	0.004031
DropConnect	0.9446	0.3533	0.1281	0.1813	0.016953
Proposed	0.9399	**0.3048**	**0.1083**	**0.1203**	**0.000053**

evaluates probabilistic regression quality (Bishop, 2006). CE90 and MPIW90 evaluate interval calibration and sharpness, respectively (Khosravi et al., 2011; Kuleshov et al., 2018). Detailed metric definitions are provided Section A1.3 of Appendix.

5 Results and Discussion

5.1 Classification Task

Table 1 reports the Fashion-MNIST classification results for VGG16 and ResNet18 across two attenuation/drop ratios, $\rho = 0.1$, and $\rho = 0.3$. Overall, the proposed Sign-Entropy-guided attenuation method gives the best probabilistic performance in every architecture/drop-ratio block. Specifically, it obtains the lowest NLL, Brier score, AECE, and epistemic uncertainty across all reported VGG16 and ResNet18 settings. This indicates that the proposed method consistently improves the quality of the predictive distribution and calibration-related metrics compared with Plain BNN, Dropout, DropConnect, and Concrete Dropout.

In terms of classification accuracy, the proposed method preserves comparable macro-F1. For VGG16, it achieves the best F1 at $\rho = 0.1$, and is comparable to Concrete Dropout at $\rho = 0.3$ while giving substantially better NLL, Brier score, AECE, and epistemic uncertainty. For ResNet18, DropConnect or Concrete Dropout sometimes achieves slightly higher F1, but this comes at the cost of higher calibration error and epistemic uncertainty. For example, at $\rho = 0.3$, Concrete Dropout gives the highest F1, whereas the proposed method gives lower NLL, Brier score, AECE, and epistemic uncertainty.

These results suggest that the proposed method is not merely improving point prediction accuracy, but is mainly improving the predictive–uncertainty trade-off. Random regularizers such as Dropout and DropConnect can maintain or improve F1 in some cases, but they do not consistently improve calibration and uncertainty. In contrast, Sign-Entropy-guided attenuation selectively suppresses directionally unstable posterior weights and thereby produces more reliable predictive distributions while maintaining comparable classification accuracy.

5.2 REGRESSION TASK

Table 2 reports the UTKFace age-regression results for VGG16 and ResNet18 at drop ratios $\rho = 0.1$ and $\rho = 0.3$. Overall, the proposed Sign-Entropy-guided attenuation method maintains competitive predictive accuracy while decreasing calibration error and uncertainty-related metrics. In terms of MAE, the proposed method is either the best or close to the best method across the reported settings. For VGG16, it achieves the best MAE at $\rho = 0.3$ and remains close to Concrete Dropout at $\rho = 0.1$. For ResNet18, it achieves the best MAE at both $\rho = 0.1$ and $\rho = 0.3$.

The improvement is more consistent in the calibration metrics. For VGG16, the proposed method gives the lowest NLL and age-bin Brier score at both drop ratios, indicating better predictive likelihood and better age-bin probability quality. It also gives the sharpest 90% prediction intervals, as measured by MPIW90, at both $\rho = 0.1$ and $\rho = 0.3$. At $\rho = 0.3$, the proposed method also achieves the best CE90 and the lowest epistemic uncertainty among the compared methods. Although Concrete Dropout gives lower epistemic uncertainty at $\rho = 0.1$, the proposed method gives better NLL, Brier score, and MPIW90, suggesting a stronger overall uncertainty–accuracy trade-off.

For ResNet18, the proposed method again provides a strong trade-off. At $\rho = 0.1$, it gives the best MAE and the lowest epistemic uncertainty, while its NLL and age-bin Brier score remain close to the best Concrete Dropout values. At $\rho = 0.3$, the proposed method gives the best MAE, NLL, age-bin Brier score, and MPIW90. While Plain BNN gives the lowest CE90 in this setting, the proposed method produces sharper intervals and better predictive likelihood. DropConnect is consistently unstable in regression, showing substantially worse MAE, NLL, Brier score, MPIW90, and epistemic uncertainty in most settings. Detailed results for the regression task can be found in Section A1.4 of the Appendix.

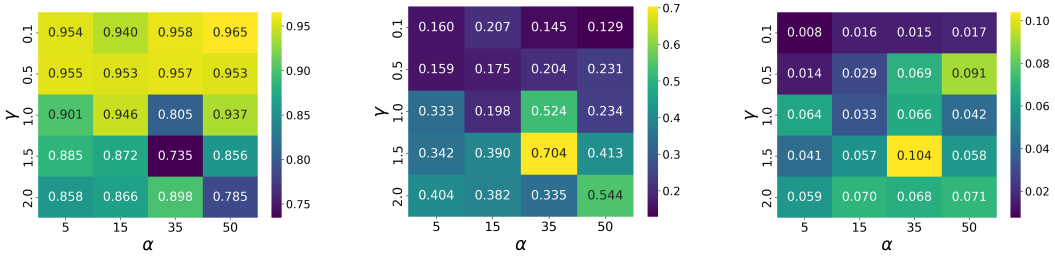
These results support the same conclusion as the classification experiments: Sign-Entropy-guided attenuation improves the quality of calibration while preserving predictive accuracy. In regression, this appears as lower NLL and age-bin Brier score, sharper predictive intervals, and reduced epistemic uncertainty where available. The proposed method therefore provides a favorable accuracy–calibration–uncertainty trade-off across both classification and regression tasks.

Table 2: UTKFace age-regression performance for VGG16 and ResNet18 split-head models at drop ratios $\rho = 0.1$ and $\rho = 0.3$. Age-bin Brier denotes the auxiliary age-bin Brier score. CE90 denotes absolute coverage error at the 90% predictive interval, and MPIW90 denotes mean prediction interval width at the same nominal coverage level. Epistemic uncertainty is denoted by Epistemic U. Lower values are better for all scores except MAE where lower also indicates better point prediction accuracy. Best values within each architecture and drop ratio block are highlighted in bold.

Method	MAE ↓	NLL ↓	Age-bin Brier ↓	CE90 ↓	MPIW90 ↓	Epistemic U ↓
VGG16, drop ratio $\rho = 0.1$						
Plain BNN	5.0285	3.3178	0.5898	**0.0137**	23.1200	3.3236
Dropout	5.0293	3.3558	0.6015	0.0173	23.5821	3.1974
DropConnect	6.3663	4.1652	0.8020	0.0986	87.4031	733.5781
Concrete Dropout	**4.8521**	3.2777	0.5726	0.0213	23.8800	**0.1194**
Proposed ($\alpha = 5, \gamma = 0.1$)	4.9628	**3.2498**	**0.5500**	0.0201	**23.0355**	0.1722
VGG16, drop ratio $\rho = 0.3$						
Plain BNN	5.4318	3.3898	0.6211	0.0151	24.7576	11.7718
Dropout	5.3743	3.3806	0.6168	0.0274	25.4865	8.2547
DropConnect	6.2360	4.0179	0.7842	0.0961	63.1557	363.8194
Concrete Dropout	5.3543	3.3166	0.5954	0.0128	24.0485	0.5423
Proposed ($\alpha = 3, \gamma = 1$)	**5.1179**	**3.2996**	**0.5917**	**0.0111**	**23.0512**	**0.3370**
ResNet18, drop ratio $\rho = 0.1$						
Plain BNN	5.4625	3.3933	0.5977	0.0072	23.5331	19.2406
Dropout	5.3100	3.3857	0.5999	**0.0021**	**23.4045**	17.1638
DropConnect	9.3724	4.5848	0.8853	0.1000	124.9823	1563.4625
Concrete Dropout	5.2404	**3.2476**	**0.5643**	0.0190	25.8869	1.0489
Proposed ($\alpha = 50, \gamma = 0.1$)	**5.2148**	3.2638	0.5681	0.0145	23.5218	**0.4223**
ResNet18, drop ratio $\rho = 0.3$						
Plain BNN	5.2918	3.3667	0.5932	**0.0021**	22.7362	17.284391
Dropout	5.3303	3.3681	0.6024	0.0109	23.8727	22.7491
DropConnect	5.5811	4.3767	0.8323	0.0994	101.5046	1249.5874
Concrete Dropout	5.2609	3.2445	0.5654	0.0094	23.6263	3.4934
Proposed ($\alpha = 50, \gamma = 0.1$)	**5.1922**	**3.2437**	**0.5564**	0.0201	**22.6336**	**3.2453**

5.3 Ablation Study

5.3.1 Hyper-parameter Sensitivity



(a) F1 score (b) NLL (c) AECE

Figure 1: Hyperparameter sensitivity of the proposed method with respect to the gate sharpness parameter α and annealing parameter γ for Resnet18 backbone model with $\rho = 0.3$. Results are shown for F1 score, NLL, and AECE. Please refer to Section A1.5 in Appendix for all results.

We perform a hyperparameter sensitivity analysis by varying the gate sharpness parameter α and the annealing parameter γ, with the attenuation ratio fixed at $\rho = 0.3$, on Fashion-MNIST using the ResNet18 backbone. The resulting heatmaps for F1 score, NLL, and AECE are shown in Figure 1.

The results show that performance is most stable for small values of γ, particularly $\gamma = 0.1$ and $\gamma = 0.5$. F1 remains high in this region, with the best value obtained at high α and low γ. The probabilistic metrics show a similar trend: NLL, Brier score, and AECE generally attain their best values when γ is small. In contrast, larger values of γ often degrade both classification performance and calibration quality, especially for intermediate and high α values.

These results suggest that the proposed attenuation mechanism benefits from a gradual annealing schedule. The parameter α controls the sharpness of the Sign-Entropy gate, while γ controls how strongly the attenuation is introduced during training. A low γ appears to avoid overly aggressive early attenuation, leading to a better accuracy–calibration trade-off. Overall, the sensitivity analysis supports treating (α, γ) as validation-selected regularization hyperparameters.

5.3.2 Posterior Mean ($\widetilde{\mu}$) and Standard Deviation($\widetilde{\sigma}$) Ablation

The results from Table 3 show that joint $(\mu + \sigma)$ attenuation provides a stronger overall trade-off than μ-only attenuation. Across the reported settings, the proposed method consistently improves F1, NLL, and Brier score, indicating that attenuating only the posterior mean is not sufficient to preserve both accuracy and achieve lower calibration error. Although μ only attenuation occasionally gives slightly lower ECE or AECE, these gains are small and do not translate into better overall predictive performance.

Overall, the ablation supports the proposed joint design. Attenuating μ reduces the expected contribution of high-Sign-Entropy weights, while attenuating both μ and σ controls their posterior spread. Joint attenuation therefore provides a more principled mechanism for limiting the influence of directionally unstable weights, leading to better accuracy and calibration across architectures and attenuation ratios. We exclude the σ-only variant because attenuating standard deviation without attenuating the mean forces the model to be more confident on the unstable weights.

6 Related Work

Dropout (Srivastava et al., 2014) is one of the most widely used stochastic regularization methods. It randomly sets activations to zero during training, discouraging co-adaptation among hidden units and improving generalization. DropConnect (Wan et al., 2013) applies a related idea at the weight level by randomly masking weights during training. Both methods introduce stochasticity into training. However, the masking is random and does not depend on whether a particular activation or weight is uncertain, stable, or directionally reliable.

Table 3: Ablation of the Sign-Entropy attenuation mechanism for VGG16 and ResNet18 classification on the Fashion-MNIST dataset. Checked boxes indicate attenuation of the corresponding posterior parameter: mean (μ) or standard deviation (σ). Attenuating both parameters corresponds to the proposed method, while attenuating only μ corresponds to the μ-only variant. Higher F1 is better; lower NLL, Brier, ECE, AECE, and epistemic uncertainty (denoted by Epistemic U) are better. Best values within each architecture/drop-ratio block are shown in bold.

Drop ratio (ρ)	μ	σ	F1 ↑	NLL ↓	Brier ↓	ECE ↓	AECE ↓	Epistemic U ↓
VGG16								
0.1	✓	✓	**0.9213**	**0.3804**	**0.1475**	**0.1351**	**0.1627**	$\mathbf{7.2\times10^{-5}}$
0.1	✓	✗	0.9083	0.4169	0.1657	0.1372	0.1644	1.123×10^{-3}
0.2	✓	✓	**0.9192**	**0.3967**	**0.1538**	0.1443	0.1748	$\mathbf{5.6\times10^{-5}}$
0.2	✓	✗	0.9102	0.4221	0.1649	**0.1393**	**0.1743**	1.008×10^{-3}
0.3	✓	✓	**0.9250**	**0.3763**	**0.1438**	**0.1330**	**0.1656**	$\mathbf{5.0\times10^{-5}}$
0.3	✓	✗	0.9118	0.4123	0.1619	0.1386	0.1693	1.078×10^{-3}
0.4	✓	✓	**0.9150**	**0.3840**	**0.1511**	0.1351	**0.1529**	$\mathbf{5.7\times10^{-5}}$
0.4	✓	✗	0.9086	0.4038	0.1609	**0.1268**	0.1536	1.135×10^{-3}
0.5	✓	✓	**0.9194**	**0.3867**	**0.1489**	**0.1334**	**0.1633**	$\mathbf{4.5\times10^{-5}}$
0.5	✓	✗	0.9075	0.4373	0.1732	0.1397	0.1830	1.146×10^{-3}
ResNet18								
0.1	✓	✓	**0.9500**	**0.2917**	**0.0979**	**0.1291**	**0.1439**	$\mathbf{4.0\times10^{-5}}$
0.1	✓	✗	0.9469	0.3075	0.1068	0.1300	0.1538	8.00×10^{-4}
0.2	✓	✓	**0.9521**	**0.2915**	**0.0975**	0.1286	0.1482	$\mathbf{3.0\times10^{-5}}$
0.2	✓	✗	0.9473	0.3012	0.1033	**0.1188**	**0.1481**	7.60×10^{-4}
0.3	✓	✓	**0.9491**	**0.2865**	**0.0986**	0.1148	**0.1391**	$\mathbf{2.9\times10^{-5}}$
0.3	✓	✗	0.9449	0.3106	0.1078	**0.1141**	0.1518	8.39×10^{-4}
0.4	✓	✓	**0.9519**	**0.2761**	**0.0937**	0.1260	**0.1328**	$\mathbf{2.6\times10^{-5}}$
0.4	✓	✗	0.9475	0.3046	0.1056	**0.1213**	0.1518	8.03×10^{-4}
0.5	✓	✓	**0.9516**	**0.2930**	**0.0995**	0.1396	0.1522	$\mathbf{3.0\times10^{-5}}$
0.5	✓	✗	0.9480	0.2968	0.1019	**0.1302**	**0.1430**	7.85×10^{-4}

Bayesian neural networks maintain distributions over weights rather than point estimates, providing a principled framework for modeling epistemic uncertainty (Blundell et al., 2015; Gal & Ghahramani, 2016). In variational BNNs, each weight is represented by a posterior mean and variance, allowing uncertainty to be propagated to predictions through Monte Carlo sampling. Most BNN approaches use posterior uncertainty mainly to estimate predictive uncertainty.

Concrete Dropout (Gal et al., 2017) learns dropout probabilities by relaxing the Bernoulli mask using the Concrete distribution. This provides an adaptive dropout mechanism and is closely connected to variational Bayesian inference. Other approaches, such as deep ensembles (Lakshminarayanan et al., 2017) and uncertainty estimation frameworks for deep learning (Loquercio et al., 2020), focus primarily on estimating predictive uncertainty. These methods are useful for quantifying uncertainty at test time, but they do not directly define a weight-level train-time regularization mechanism.

The proposed method lies between stochastic regularization and Bayesian uncertainty modeling. Like Dropout and DropConnect, it acts during training as a regularizer and like BNN methods, it uses posterior information over weights. The proposed method, unlike the above prior works, is both deterministic and uncertainty-aware alternative.

7 Conclusion

In this work, we introduced Sign Entropy based regularization that attenuates weights based on their sign instability during training. Unlike traditional Dropout and DropConnect, which apply uniform stochasticity to activations or weights, our method leverages the Bayesian posterior distribution of weights to compute Sign Entropy as a principled measure of directional uncertainty of weight influence. By prioritizing the attenuation of weights with high Sign Entropy, our method reduces the influence of directionally unstable posterior weights, leading to improved calibration. Empirical evaluations across both classification and regression tasks, show that Sign Entropy guided attenuation generally preserves predictive accuracy while decreasing calibration error and epistemic uncertainty. This approach provides a targeted, uncertainty-aware form of regularization and opens up new directions for entropy-guided network pruning, adaptive sparsity, and robust Bayesian deep learning.

REFERENCES

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Revoti Prasad Bora, Philipp Terhörst, Raymond Veldhuis, Raghavendra Ramachandra, and Kiran Raja. CoFE: Consistency-Driven Feature Elimination for eXplainable AI. In *International Conference on Pattern Recognition*, pp. 368–382. Springer, 2025a.

Revoti Prasad Bora, Philipp Terhörst, Raymond Veldhuis, Raghavendra Ramachandra, and Kiran Raja. Bay-cofe: Bayesian consistency-driven feature elimination for explainable ai. *Neurocomputing*, pp. 132219, 2025b.

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

Tianfeng Chai and Roland R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? arguments against avoiding rmse in the literature. *Geoscientific Model Development*, 7(3): 1247–1250, 2014.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. *Advances in neural information processing systems*, 30, 2017.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1321–1330. JMLR.org, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Mate Lengyel. Bayesian active learning for classification and preference learning. In *arXiv preprint arXiv:1112.5745*, 2011.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Abbas Khosravi, Saeid Nahavandi, Doug Creighton, and Amir F. Atiya. Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Transactions on Neural Networks*, 22(9):1341–1356, 2011.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, 2018.

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

Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, 2020. doi: 10.1109/LRA.2020.2974682.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet K. Dokania. Calibrating deep neural networks using focal loss. In *Advances in Neural Information Processing Systems*, 2020.

Gary W Oehlert. A note on the delta method. *The American Statistician*, 46(1):27–29, 1992.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, volume 15, pp. 1929–1958, 2014.

C. J. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 1979.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

A1 APPENDIX

A1.1 THEORETICAL MOTIVATION

Setting. Let x denote an input sample and let $h(x) \in \mathbb{R}^d$ denote the feature vector produced by the deterministic backbone before the final Bayesian layer. We write $h_j(x)$ for the j-th feature activation. In our method, stochasticity is restricted to the final Bayesian layer. Each final-layer weight has a Gaussian variational posterior

$$w_j \sim \mathcal{N}(\mu_j, \sigma_j^2) \tag{20}$$

For binary classification, the final logit can be written as

$$z(x) = \sum_{j=1}^{d} w_j h_j(x) + b, \tag{21}$$

where $z(x)$ is the scalar logit, $h_j(x)$ is the j-th feature activation, w_j is the corresponding final-layer Bayesian weight, and b is the bias term. The predicted probability is

$$\hat{p}(x) = \mathrm{sig}(z(x)) = \frac{1}{1 + \exp(-z(x))}, \tag{22}$$

where $\mathrm{sig}(\cdot)$ is the sigmoid function. The predicted class is

$$\hat{Y} = \mathbf{1}\{z(x) > 0\}, \tag{23}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. Thus, the sign of the logit determines the predicted class.

Sign Entropy. For each Bayesian weight $w_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$, we compute the posterior probability that the weight is positive or negative:

$$p_j^+ = \mathbb{P}(w_j > 0) = \Phi\left(\frac{\mu_j}{\sigma_j + \epsilon}\right), \qquad p_j^- = \mathbb{P}(w_j < 0) = \Phi\left(-\frac{\mu_j}{\sigma_j + \epsilon}\right), \tag{24}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, ϵ is a small numerical constant, p_j^+ is the probability that weight j is positive, and p_j^- is the probability that weight j is negative.

The Sign Entropy (Bora et al., 2025a) of weight j is defined as:

$$H_s(w_j) = -p_j^+ \log_2 p_j^+ - p_j^- \log_2 p_j^- . \tag{25}$$

Here, $H_s(w_j)$ measures uncertainty in the *direction* of the feature effect. If $H_s(w_j)$ is close to zero, the posterior assigns almost all probability to one sign, so the feature effect has a stable direction. If $H_s(w_j)$ is close to one bit, then $p_j^+ \approx p_j^- \approx 0.5$, so the model is uncertain whether the corresponding feature contributes positive or negative contribution.

Single-feature case: Sign Entropy equals predictive class entropy. Consider the simplest case with one feature, no bias, and a positive activation (i.e., $h(x) > 0$). The logit '$z(x)$' is defined as:

$$z(x) = wh(x). \tag{26}$$

Since $h(x) > 0$, the sign of the logit is exactly the sign of the weight:

$$\mathrm{sign}(z(x)) = \mathrm{sign}(w). \tag{27}$$

Therefore,

$$\hat{Y} = \mathbf{1}\{z(x) > 0\} = \mathbf{1}\{w > 0\}. \tag{28}$$

Hence, under the posterior over w, the predicted class is a Bernoulli random variable:

$$\mathbb{P}(\hat{Y} = 1 \mid x) = p^+, \qquad \mathbb{P}(\hat{Y} = 0 \mid x) = p^-. \tag{29}$$

The entropy of the predicted class is therefore

$$H(\hat{Y} \mid x) = -p^+ \log_2 p^+ - p^- \log_2 p^- = H_s(w). \tag{30}$$

Thus, in the single-feature case, Sign Entropy is not merely a heuristic. It is exactly the posterior entropy of the predicted class. This also explains why the criterion is activation-free in this case: multiplying the weight by a positive activation scales both the posterior mean and posterior standard deviation of the contribution by the same factor, and therefore does not change the sign probability.

Multi-feature extension. For multiple features, the binary logit is

$$z(x) = \sum_{j=1}^{d} w_j h_j(x) + b. \tag{31}$$

Since the final-layer weights are Gaussian and mean-field independent, the logit is also Gaussian at fixed x (Bishop, 2006), (Blundell et al., 2015) and is given as:

$$z(x) \mid x \sim \mathcal{N}\left(m(x), s^2(x)\right), \tag{32}$$

with

$$m(x) = \sum_{j=1}^{d} \mu_j h_j(x) + b, \qquad s^2(x) = \sum_{j=1}^{d} \sigma_j^2 h_j^2(x). \tag{33}$$

Here, $m(x)$ is the posterior mean of the logit and $s^2(x)$ is the posterior variance of the logit. The term $\mu_j h_j(x)$ is the mean contribution of feature j to the logit, while $\sigma_j^2 h_j^2(x)$ is its contribution to logit variance.

The posterior probability that the predicted class is positive is

$$\mathbb{P}(\hat{Y} = 1 \mid x) = \mathbb{P}(z(x) > 0 \mid x) = \Phi\left(\frac{m(x)}{s(x)}\right). \tag{34}$$

Therefore, the entropy of the predicted class is

$$H(\hat{Y} \mid x) = H_s\left(\Phi\left(\frac{|m(x)|}{s(x)}\right)\right), \tag{35}$$

Equation (35) shows that predictive class uncertainty is controlled by the logit-level signal-to-noise ratio $|m(x)|/s(x)$. High uncertainty arises when the posterior logit distribution places substantial mass on both sides of the decision boundary.

The per-weight Sign Entropy $H_s(w_j)$ is an input-independent proxy for this logit-level uncertainty. A high-Sign-Entropy weight is directionally unstable: under posterior sampling it can switch between positive and negative contribution. If many such weights contribute to the final logit, the sign of the logit can become unstable, especially near the decision boundary.

From logit variation to confidence variation. The predicted confidence is determined by the sigmoid of the logit. Since

$$\mathrm{sig}'(z) = \mathrm{sig}(z)\,(1 - \mathrm{sig}(z)) \leq \frac{1}{4}, \tag{36}$$

we have the global Lipschitz bound

$$|\mathrm{sig}(z) - \mathrm{sig}(z')| \leq \frac{1}{4}|z - z'|. \tag{37}$$

This means that variation in the logit induces variation in predicted confidence. The effect is strongest near the decision boundary, where $z \approx 0$ and the sigmoid slope is largest.

For small logit variance, a local delta-method approximation (Oehlert, 1992) gives

$$\mathrm{Var}(\hat{p} \mid x) \approx \left[\mathrm{sig}(m(x))\,(1 - \mathrm{sig}(m(x)))\right]^2 s^2(x). \tag{38}$$

Here, $\mathrm{Var}(\hat{p} \mid x)$ is the posterior variance of the predicted probability, $m(x)$ is the posterior mean logit, and $s^2(x)$ is the posterior logit variance. This approximation shows that logit variance is most relevant for confidence variation when the sample lies near the decision boundary.

When prediction is performed by Monte Carlo marginalization, posterior class instability directly affects the reported predictive distribution. If high sign entropy weights cause different posterior samples to favor different classes, the averaged predictive probability becomes less stable and the confidence assigned to the final prediction changes. Since calibration compares predicted confidence with empirical correctness, attenuating such unstable posterior class fluctuations provides a direct mechanism by which sign entropy guided attenuation can improve calibration. This mechanism links sign entropy to calibration-relevant predictive instability, which we quantify using NLL, Brier score, and adaptive calibration error.

Sign Entropy and its relation to Signal to Noise Ratio Sign Entropy is related to the posterior signal-to-noise ratio (Blundell et al., 2015)

$$\mathrm{SNR}_j = \frac{|\mu_j|}{\sigma_j}, \tag{39}$$

where SNR_j measures the magnitude of the posterior mean relative to posterior uncertainty. For a Gaussian posterior, ranking weights by descending Sign Entropy is equivalent to ranking them by ascending SNR. However, Sign Entropy gives a bounded and probabilistic parameterization of this uncertainty. It lies in $[0, 1]$ bits, is unit-free, and directly measures the entropy of the posterior sign variable. Moreover, away from numerical boundary cases, it is a smooth function of the posterior parameters. This bounded entropy scale is useful for defining a stable train-time attenuation gate.

Additionally, our contribution is not post-training SNR pruning. Classical SNR pruning identifies low-SNR weights after training and removes them as a hard compression step. In contrast, our method uses Sign Entropy during training to softly attenuate uncertain posterior parameters. Therefore, it is closer in spirit to train-time regularizers such as Dropout and DropConnect. Dropout randomly suppresses activations; DropConnect randomly suppresses weights; our method deterministically attenuates weights whose posterior sign is uncertain.

A1.2 Implementation Details

During training, Sign Entropy is recomputed from the current posterior parameters of the final Bayesian layer. The high-entropy weights are selected using the quantile threshold, and their posterior parameters are attenuated. This operation is applied during training and is not a post-training pruning step. At inference, no additional random pruning mask is sampled; the final learned posterior parameters are used for prediction.

The model is trained with the same predictive objective as the corresponding baselines. For classification, we use cross-entropy with label smoothing. When CutMix is used, the loss for a mixed sample is

$$\mathcal{L}_{\mathrm{cls}} = \lambda\,\mathrm{CE}(f_\theta(x), y_a) + (1-\lambda)\,\mathrm{CE}(f_\theta(x), y_b), \tag{40}$$

where $f_\theta(x)$ denotes the model logits, y_a and y_b are the two CutMix labels, and λ is the CutMix mixing coefficient. The proposed method modifies the final-layer posterior parameters used by the model during training; it does not introduce a separate post-hoc pruning objective.

Extension to multi-class classification and regression. For K-class classification, the final layer contains one Bayesian weight vector per class. The class logits are

$$z_k(x) = \sum_{j=1}^{d} w_{kj} h_j(x) + b_k, \qquad k = 1, \ldots, K, \tag{41}$$

where $z_k(x)$ is the logit for class k, w_{kj} is the Bayesian weight connecting feature j to class k, $h_j(x)$ is the j-th feature activation, and b_k is the class-specific bias. Sign Entropy is computed entrywise for the final-layer weight matrix, and the quantile threshold is computed over the flattened final-layer weights.

For regression, there is no class flip. Instead, the sign of a weight determines whether a feature increases or decreases the predicted mean. High Sign Entropy therefore indicates uncertainty in the direction of the feature effect on the predicted value. In regression experiments, we evaluate predictive quality and uncertainty using regression-appropriate metrics such as MAE, RMSE, NLL, and predictive interval coverage.

Relation to Dropout, DropConnect, and SNR pruning. Dropout randomly suppresses activations during training, while DropConnect randomly suppresses weights during training. In contrast, the proposed method deterministically attenuates weights whose posterior sign is uncertain. Although Sign Entropy is monotone-related to SNR for Gaussian posteriors, our method is different from classical SNR-based pruning. SNR pruning is typically applied after training as a hard model-compression step. The proposed method is instead a train-time, soft, annealed attenuation mechanism. Thus, it is experimentally comparable to train-time regularizers such as Dropout and DropConnect, but its selection is guided by posterior directional uncertainty rather than random masking. A pictorial representation of the proposed approach is provided in Figure A1.

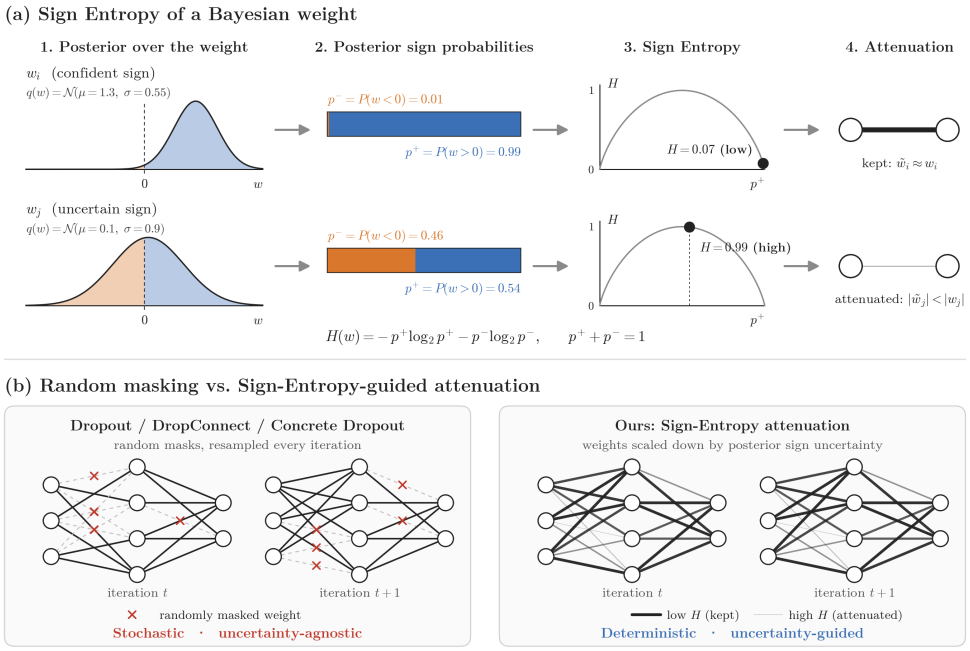


Figure A1: Overview of the proposed Sign Entropy guided attenuation mechanism. (a) Sign Entropy quantifies the directional uncertainty of a Bayesian weight from its posterior sign probabilities. A weight with a confident posterior sign has low Sign Entropy and is largely preserved, whereas a weight with an uncertain sign has high Sign Entropy and is attenuated. (b) Comparison with stochastic masking methods such as Dropout, DropConnect, and Concrete Dropout. While these methods randomly mask weights across iterations without considering their uncertainty, the proposed approach deterministically scales weights according to posterior sign uncertainty, preserving low-entropy weights and suppressing high-entropy weights.

A1.3 Details of Evaluation Metrics

For classification, we evaluate predictive performance using macro-F1 score, which averages the F1 score across classes and is appropriate when class-wise performance is important (van Rijsbergen, 1979). We evaluate probabilistic prediction quality using negative log-likelihood (NLL) (Bishop, 2006), the multiclass Brier score (Brier, 1950), and Adaptive Expected Calibration Error (AECE) (Mukhoti et al., 2020). NLL measures the log-loss of the predictive distribution, Brier score measures the squared error between predicted probabilities and one-hot labels, and AECE estimates calibration error using adaptive/equal-mass confidence bins.

For each test input, we perform S Monte Carlo forward passes through the Bayesian final layer. Let $p^{(s)}(y = c \mid x)$ denote the predicted probability of class c at Monte Carlo sample s, and let

$$\bar{p}_c(x) = \frac{1}{S} \sum_{s=1}^{S} p^{(s)}(y = c \mid x) \tag{42}$$

be the Monte Carlo predictive probability for class c. Classification NLL is computed from the predictive probability assigned to the true class:

$$\mathrm{NLL} = -\frac{1}{N} \sum_{i=1}^{N} \log \bar{p}_{y_i}(x_i), \tag{43}$$

where N is the number of test samples and y_i is the true label of sample i (Bishop, 2006).

The multiclass Brier score is computed as

$$\mathrm{Brier} = \frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \left(\bar{p}_c(x_i) - \mathbf{1}\{y_i = c\} \right)^2, \tag{44}$$

where C is the number of classes and $\mathbf{1}\{y_i = c\}$ is the one-hot class indicator (Brier, 1950). Calibration is evaluated using AECE, which compares average confidence and empirical accuracy in adaptive confidence bins (Mukhoti et al., 2020).

We also report epistemic uncertainty using the mutual-information/BALD-style decomposition of predictive uncertainty (Houlsby et al., 2011; Kendall & Gal, 2017). The total predictive uncertainty is the entropy of the Monte Carlo averaged predictive distribution:

$$\mathcal{H}_{\text{total}}(x) = -\sum_{c=1}^{C} \bar{p}_c(x) \log \bar{p}_c(x). \tag{45}$$

The expected data uncertainty is

$$\mathcal{H}_{\text{alea}}(x) = \frac{1}{S} \sum_{s=1}^{S} \left[-\sum_{c=1}^{C} p^{(s)}(y = c \mid x) \log p^{(s)}(y = c \mid x) \right]. \tag{46}$$

The epistemic uncertainty is then

$$\mathcal{H}_{\text{epi}}(x) = \mathcal{H}_{\text{total}}(x) - \mathcal{H}_{\text{alea}}(x). \tag{47}$$

In the main classification results, we report the mean epistemic uncertainty over the test set.

For regression, we evaluate predictive accuracy using mean absolute error (MAE) and root mean squared error (RMSE) (Chai & Draxler, 2014). If $\hat{y}_i$ is the predicted mean and y_i is the true target, then

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|, \tag{48}$$

and

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}. \tag{49}$$

We evaluate probabilistic regression quality using Gaussian negative log-likelihood (Bishop, 2006). For a predictive Gaussian distribution with mean $\hat{\mu}_i$ and variance $\hat{\sigma}_i^2$, the regression NLL is

$$\text{NLL} = \frac{1}{N} \sum_{i=1}^{N} \left[\frac{1}{2} \log \hat{\sigma}_i^2 + \frac{(y_i - \hat{\mu}_i)^2}{2\hat{\sigma}_i^2} \right] + \text{const}. \tag{50}$$

For regression uncertainty calibration, we report predictive interval coverage error and Mean Prediction Interval Width (MPIW) at nominal coverage levels 68%, 90%, and 95% (Khosravi et al., 2011; Kuleshov et al., 2018). Let PICP_q denote the empirical predictive interval coverage at nominal level q. The absolute coverage error CE_q is defined as:

$$\text{CE}_q = |\text{PICP}_q - q|. \tag{51}$$

MPIW measures the average width or sharpness of the corresponding predictive intervals and if defined as:

$$\text{MPIW}_q = \frac{1}{N} \sum_{i=1}^{N} \left(U_i^{(q)} - L_i^{(q)} \right), \tag{52}$$

where $L_i^{(q)}$ and $U_i^{(q)}$ are the lower and upper bounds of the predictive interval for sample i at nominal coverage level q. Lower coverage error indicates better calibrated intervals, while lower MPIW indicates sharper predictive intervals when coverage is comparable.

A1.4 Additional Results Regression Task

In this sub-section we provide additional results for the regression task in tables Table A1 and Table A2.

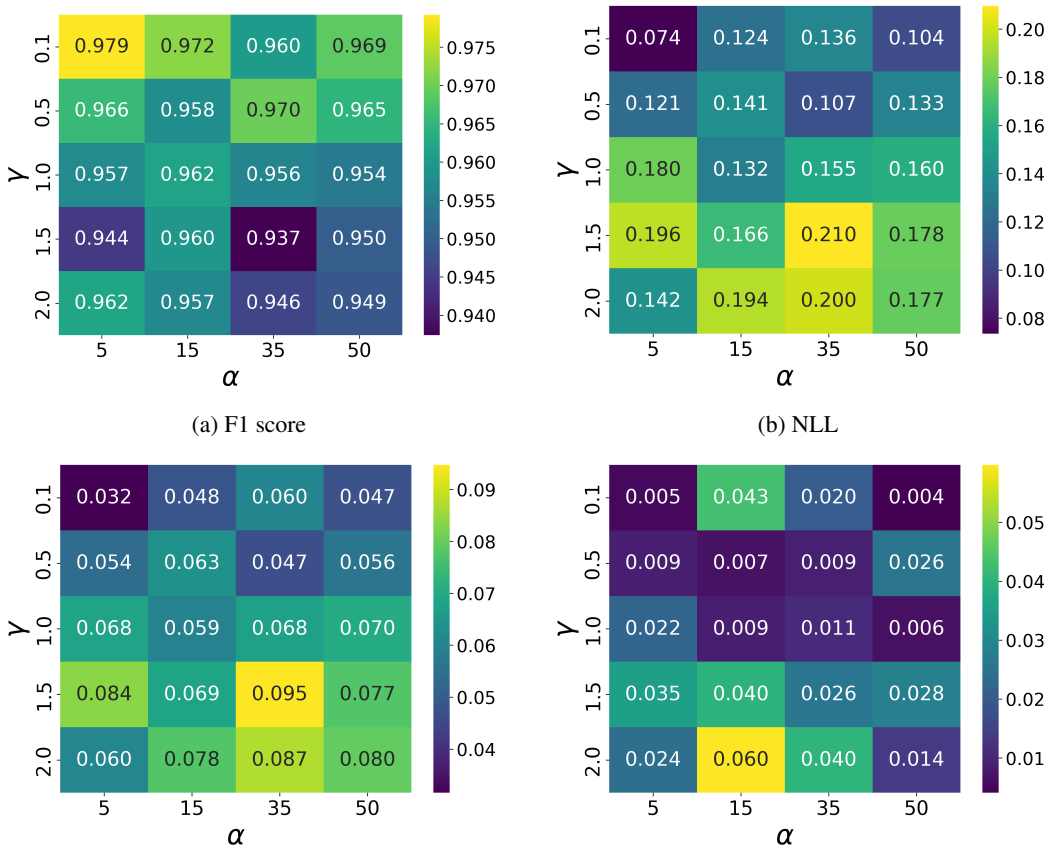
Table A1: Validation-calibrated VGG split-head age-regression performance on the UTKFace dataset for drop ratios $\rho = 0.3$ and $\rho = 0.1$. For Sign-Entropy, $\rho = 0.3$ uses $(\alpha = 3, \gamma = 1)$, while $\rho = 0.1$ uses $(\alpha = 5, \gamma = 0.1)$. CE denotes absolute coverage error and MPIW denotes mean prediction interval width. Lower values are better. Best values within each drop-ratio block are shown in bold.

Drop ratio	Method	MAE ↓	NLL ↓	Brier ↓	Aleatoric U ↓	Epistemic U ↓	CE68 ↓	CE90 ↓	CE95 ↓	MPIW68 ↓	MPIW90 ↓	MPIW95 ↓
$\rho = 0.1$	Plain BNN	5.0285	3.3178	0.5898	49.5392	3.3236	**0.0869**	**0.0137**	**0.0002**	14.0547	23.1200	27.5472
$\rho = 0.1$	Dropout	5.0293	3.3558	0.6015	50.7116	3.1974	0.0886	0.0173	0.0006	14.3356	23.5821	28.0978
$\rho = 0.1$	DropConnect	6.3663	4.1652	0.8020	**40.4352**	733.5781	0.3133	0.0986	0.0492	53.1326	87.4031	104.1399
$\rho = 0.1$	Concrete Dropout	**4.8521**	3.2777	0.5726	64.4692	**0.1194**	0.0987	0.0213	0.0030	14.5167	23.8800	28.4528
$\rho = 0.1$	Proposed $(\alpha = 5, \gamma = 0.1)$	4.9628	**3.2498**	**0.5500**	57.4442	0.1722	0.0967	0.0201	0.0039	**14.0034**	**23.0355**	**27.4466**
$\rho = 0.3$	Plain BNN	5.4318	3.3898	0.6211	47.2020	11.7718	0.0801	0.0151	0.0029	15.0502	24.7576	29.4984
$\rho = 0.3$	Dropout	5.3743	3.3806	0.6168	54.6030	8.2547	0.0945	0.0274	0.0064	15.4933	25.4865	30.3668
$\rho = 0.3$	DropConnect	6.2360	4.0179	0.7842	**17.6938**	363.8194	0.2899	0.0961	0.0478	38.3926	63.1557	75.2494
$\rho = 0.3$	Concrete Dropout	5.3543	3.3166	0.5954	62.8022	0.5423	**0.0762**	0.0128	**0.0013**	14.6192	24.0485	28.6536
$\rho = 0.3$	Proposed $(\alpha = 3, \gamma = 1)$	**5.1179**	**3.2996**	**0.5917**	55.0825	**0.3370**	0.0796	**0.0111**	0.0037	**14.0129**	**23.0512**	**27.4653**

Table A2: Validation-calibrated ResNet18 split-head age-regression performance on the UTKFace dataset for drop ratios $\rho = 0.1$ and $\rho = 0.3$. CE denotes absolute coverage error and MPIW denotes Mean Prediction Interval Width. Lower values are better. Best available values for $\rho = 0.3$ are shown in bold.

Drop ratio	Method	MAE ↓	NLL ↓	Brier ↓	Aleatoric U ↓	Epistemic U	CE68 ↓	CE90 ↓	CE95 ↓	MPIW68 ↓	MPIW90 ↓	MPIW95 ↓
$\rho = 0.1$	Plain BNN	5.4625	3.3933	0.5977	**34.2833**	19.2406	**0.0596**	0.0072	0.0062	14.3058	23.5331	28.0394
$\rho = 0.1$	Dropout	5.3100	3.3857	0.5999	35.5520	17.1638	0.0652	**0.0021**	0.0085	**14.2276**	**23.4045**	**27.8862**
$\rho = 0.1$	DropConnect	9.3724	4.5848	0.8853	105.1146	1563.4625	0.3189	0.1000	0.0500	75.9771	124.9823	148.9151
$\rho = 0.1$	Concrete Dropout	5.2404	**3.2476**	**0.5643**	87.3973	1.0489	0.0767	0.0190	**0.0011**	15.7367	25.8869	30.8439
$\rho = 0.1$	Proposed ($\alpha = 50, \gamma = 0.1$)	**5.2148**	3.2638	0.5681	62.7318	**0.4223**	0.0742	0.0145	0.0037	14.2990	23.5218	28.0260
$\rho = 0.3$	Plain BNN	5.2918	3.3667	0.5932	**34.0238**	17.284391	**0.0576**	**0.0021**	0.0138	13.8214	22.7362	27.0900
$\rho = 0.3$	Dropout	5.3303	3.3681	0.6024	50.8260	22.7491	0.0680	0.0109	0.0076	14.5123	23.8727	28.4441
$\rho = 0.3$	DropConnect	5.5811	4.3767	0.8323	89.0085	1249.5874	0.3169	0.0994	0.0494	61.7049	101.5046	120.9416
$\rho = 0.3$	Concrete Dropout	5.2609	3.2445	0.5654	65.0220	3.4934	0.0652	0.0094	**0.0023**	14.3625	23.6263	28.1505
$\rho = 0.3$	Proposed ($\alpha = 50, \gamma = 0.1$)	**5.1922**	**3.2437**	**0.5564**	54.3117	**3.2453**	0.0621	0.0201	0.0030	**13.7590**	**22.6336**	**26.9676**

A1.5 Comprehensive Results for Hyper-parameter sensitivity



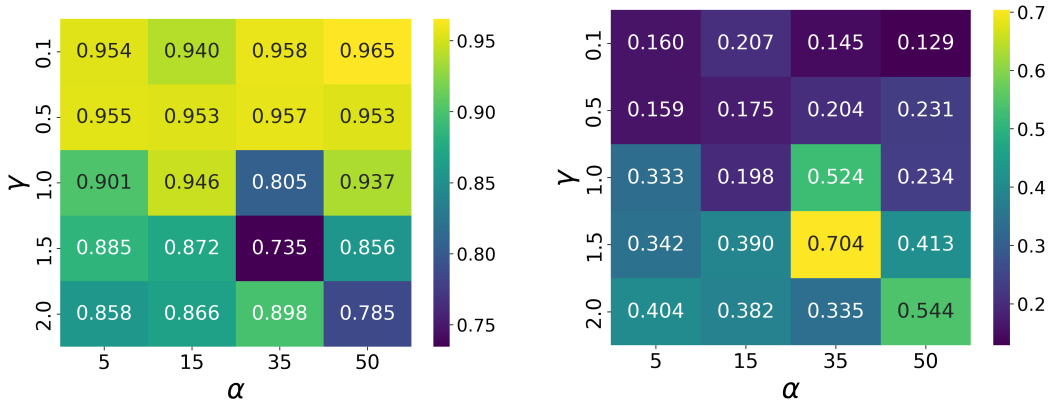
(a) F1 score

(b) NLL

(c) Brier score

(d) AECE

Figure A2: Hyperparameter sensitivity of the proposed Sign-Entropy-guided attenuation method with respect to the gate sharpness parameter α and annealing parameter γ for $\rho = 0.1$. Results are shown for F1 score, NLL, Brier score, and AECE.



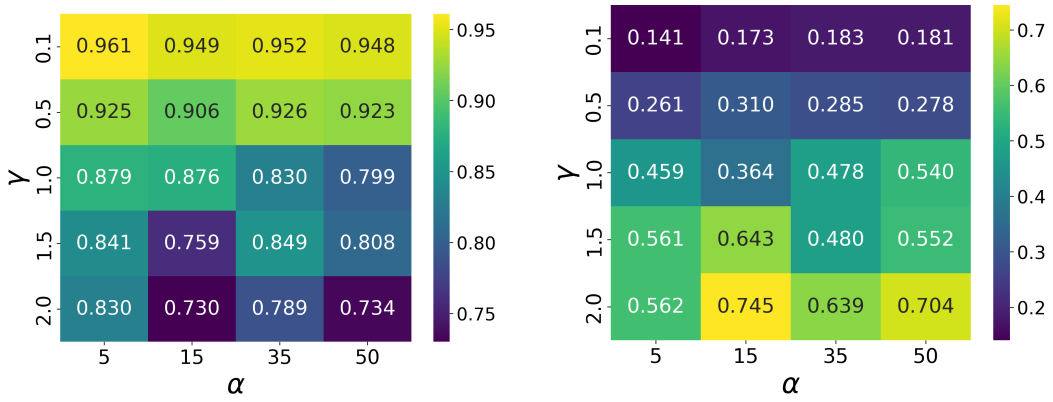
(a) F1 score

(b) NLL

(c) Brier score

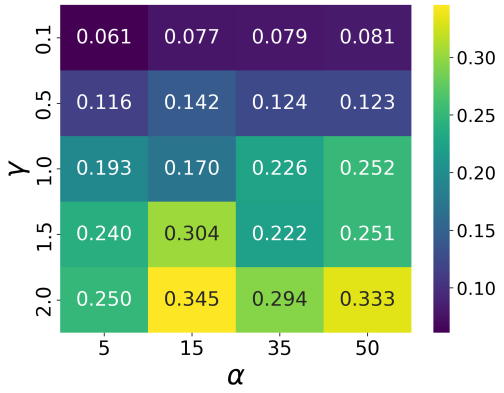
(d) AECE

Figure A3: Hyperparameter sensitivity of the proposed Sign-Entropy-guided attenuation method with respect to the gate sharpness parameter α and annealing parameter γ for $\rho = 0.3$. Results are shown for F1 score, NLL, Brier score, and AECE.

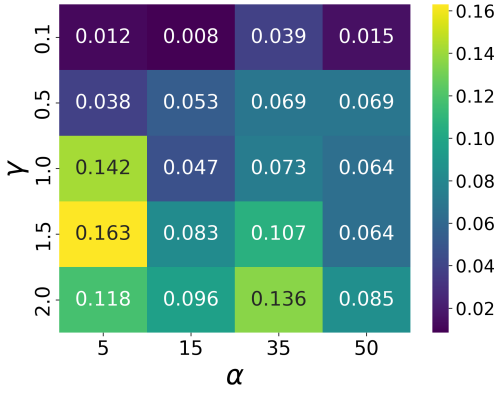


(a) F1 score

(b) NLL



(c) Brier score



(d) AECE

Figure A4: Hyperparameter sensitivity of the proposed Sign-Entropy-guided attenuation method with respect to the gate sharpness parameter α and annealing parameter γ for $\rho = 0.5$. Results are shown for F1 score, NLL, Brier score, and AECE.