A PID Controller Approach for Adaptive Probability-dependent Gradient Decay in Model Calibration

Siyuan Zhang
School of Internet of Things Engineering
Jiangnan University
Wuxi, China 214122

Linbo Xie *
School of Internet of Things Engineering
Jiangnan University
Wuxi, China 214122

Abstract

Modern deep learning models often exhibit overconfident predictions, inadequately capturing uncertainty. During model optimization, the expected calibration error tends to overfit earlier than classification accuracy, indicating distinct optimization objectives for classification error and calibration error. To ensure consistent optimization of both model accuracy and model calibration, we propose a novel method incorporating a probability-dependent gradient decay coefficient into loss function. This coefficient exhibits a strong correlation with the overall confidence level. To maintain model calibration during optimization, we utilize a proportional-integral-derivative (PID) controller to dynamically adjust this gradient decay rate, where the adjustment relies on the proposed relative calibration error feedback in each epoch, thereby preventing the model from exhibiting over-confidence or under-confidence. Within the PID control system framework, the proposed relative calibration error serves as the control system output, providing an indication of the overall confidence level, while the gradient decay rate functions as the controlled variable. Moreover, recognizing the impact of gradient amplitude of adaptive decay rates, we implement an adaptive learning rate mechanism for gradient compensation to prevent inadequate learning of over-small or over-large gradient. Empirical experiments validate the efficacy of our PID-based adaptive gradient decay rate approach, ensuring consistent optimization of model calibration and model accuracy. The code of implementation is available in https://github.com/UHIF/PID_AGD.

1 Inroduction

Model calibration aims to refine the uncertainty distribution of model output, ensuring its faithful reflection of the inherent uncertainty characteristics. Softmax mapping is commonly employed in the baseline learning to establish the output-probability mapping. Yet, with the escalation in model parameterizations, the output uncertainty distribution from over-parameterized models tends to become over-confident [1]. Consequently, the baseline strategy, which exclusively depends on Softmax mapping without accounting for calibration characteristics as optimization objectives, fails to achieve perfect model calibration. Particularly in high-risk applications, inadequate model calibration poses heightened safety risks [2, 3].

There are three primary strategies for calibrating uncertainty in deep learning models: Bayesian neural networks, post-processing calibration, and training-based model calibration. Bayesian neural networks (BNNs) integrate Bayesian inference into their framework, distinguishing them from traditional neural networks [4, 5]. In contrast to conventional neural networks, which yield point estimates,

*Linbo Xie is the corresponding author. E-mail: xie_linbo@jiangnan.edu.cn

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3642

BNNs offer a probability distribution across potential outputs, enabling uncertainty quantification in predictions [6]. This is accomplished by assigning prior distributions to the network's parameters and iteratively updating these distributions using Bayes' theorem as more data is acquired.

The post-processing calibration methods involve establishing the additional output-probability relationship through supplementary mappings to refine output uncertainty distribution [7, 8, 9]. This calibration method avoids disrupting the original model's decision-making pertaining to the primary task, maintaining the original generalization performance [10]. However, a notable drawback emerges: the necessity of additional structure to establish the mapping between outputs and probabilities [11]. This task possesses unique characteristics, demanding calibrated properties to construct output-probability mapping structures, optimization goals, and strategies. Unlike typical loss functions and metrics in machine learning, uncertainty is sample-specific; however, validating it individually is infeasible [12]. Furthermore, when validated collectively, it fails to fully represent individual sample properties, presenting challenges in calibrating individual sample. This presents a difficulty in calibrating model confidence via post-processing structures [13, 14].

Another approach to model calibration involves integrating various elements into the baseline optimization strategy for deep learning [7]. This enhancement of elements during the optimization leads to an improvement in the uncertainty distribution of the model output. These methods encompass a range of techniques including pre-training [15], data augmentation [16], label smoothing [17], weight decay [1], early stopping [18], structure sparsity [19], convolutional structure [20], and others. These methods not only bolster the calibration of the model and refine the output uncertainty distribution but also delve into the dynamic attributes of the model optimization, thereby enhancing decision-making. Additionally, they improve the interpretability in the decision-making and optimization process. Furthermore, certain loss functions for model output distribution have been devised based on uncertainty properties. These include MMCE [14], Correctness Ranking Loss [21], CALS [22], Focal loss [23, 24], and FLSD [25]. They jointly address classification accuracy and confidence calibration to mitigate the inclination of the over-confidence and under-confidence.

In the optimization of Softmax-based cross-entropy loss, model accuracy and model calibration represent distinct optimization characteristics [26]. Model calibration tends to be overfitting earlier in the optimization compared to accuracy [1]. To ensure consistent optimization between model accuracy and calibration, it is imperative to achieve high accuracy while maintaining adequate calibration. Our approach introduces a hyperparameter that controls the gradient decay rate within the Softmax output-probability mapping. It indicated a negative correlation between the gradient decay rate with increasing instance-level probability and the overall confidence distribution [27]. To achieve consistent optimization of accuracy and calibration, we propose detecting model calibration through a validation set in the optimization process. Drawing inspiration from the notable success of proportional-integral-derivative (PID) controllers in automatic control systems [28], we introduce an Adaptive Probability-dependent Gradient Decay through PID controller approach to calibrate the model. Moreover, considering that probability-dependent gradient decay rate may impact gradient amplitude, we design a dynamic learning rate mechanism corresponding to the changing gradient decay rate to offset fluctuations in gradient amplitude.

Our main contributions in this work can be summarized as follows: (1) We propose adaptive probability-dependent gradient decay via PID controller. This approach utilizes the feedback mechanism from automatic control to detect model calibration and adjust the probability-dependent gradient decay rate coefficient in Softmax, ensuring consistent optimization of model calibration and accuracy. (2) To counteract fluctuations in gradient magnitude caused by the adaptive probability-dependent gradient decay rate, we introduce a dynamic learning rate schedule to follow the dynamic decay rate. (3) Empirical experiments confirm the effectiveness of our approach, achieving improved model calibration while maintaining the model's generalization ability.

2 Problem formulation

2.1 Model calibration

Considering a dataset $\left\{(x^i, y^i)\right\}_{i=1}^{N} \subset \mathbf{R}^n \times \mathbf{R}^m$ and classifier f maps x to the outputs $z_j, j = 1, \ldots, m$ on m classes and $k = \arg\max_j z_j$. The ground-truth y and predicted labels $\hat{y}$ are formulated in one-hot format where $y_c = 1$ and $\hat{y}_k = 1$, where c represents the truth class. The associated confidence score of the predicted label in baseline is $\hat{p} = \max s_j(z), j = 1, \ldots, m$, where $s(\cdot)$

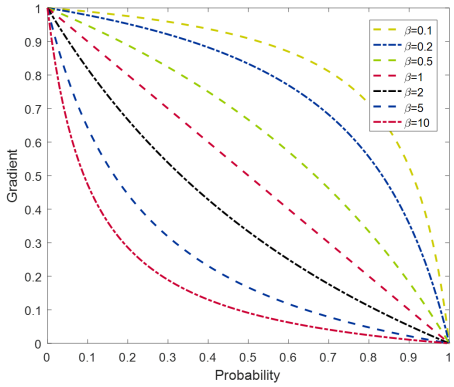


Figure 1: The gradient magnitude of different gradient decay β with increasing probability p_c.

represents Softmax mapping $R^m \to R^m$. However, Softmax mapping probabilities are not accurately reflected in the properties of model output [1]. The probability estimation of modern deep models may show over-confidence or under-confidence.

Confidence Calibration Perfect calibration of neural network can be realized when the confidence score reflects the real probability that the classification is classified correctly. Formally, the perfectly calibrated network satisfied $\mathrm{P}\left(\hat{y} = y|\hat{p} = p\right) = p$ for all $p \in [0, 1]$. However, in practical applications, the sample is divided into M bins $\{D_b\}_{b=1}^{M}$. The limited availability of data restricts the ability to accurately estimate the calibration error. According to their confidence scores and the calibration error, an approximation is calculated for each bins $\{D_b\}_{b=1}^{M}$. D_b contains all sample with $\hat{p} \in \left[\frac{b}{M}, \frac{b+1}{M}\right)$. Average confidence is computed as $conf\left(D_b\right) = \frac{1}{|D_b|}\sum_{i \in D_b} \hat{p}^i$ and the bin accuracy is computed as $acc\left(D_b\right) = \frac{1}{|D_b|}\sum_{i \in D_b} \mathrm{I}\left(y_c^i = \hat{y}_c^i\right)$. ECE and MCE [29] are calculated as follows.

$$ECE = \sum_{b=1}^{M} \frac{|D_b|}{N} \left|acc\left(D_b\right) - conf\left(D_b\right)\right| \tag{1}$$

$$MCE = \max_{b \in \{1,\ldots,M\}} \left|acc\left(D_b\right) - conf\left(D_b\right)\right| \tag{2}$$

2.2 Parametric Softmax

Softmax cross-entropy (CE) is expressed as

$$J = -\log \frac{e^{z_c}}{\sum_{j=1}^{m} e^{z_j}} \tag{3}$$

We introduce two hyperparameters in the Softmax mapping, which is expressed as follows:

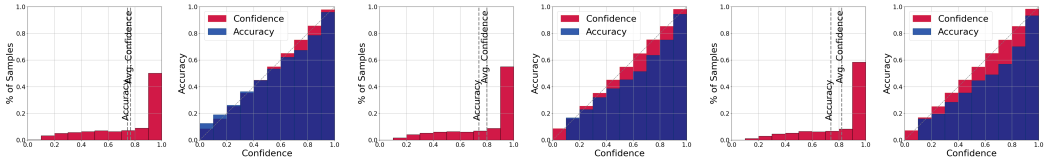
$$J = -\log \frac{e^{z_c/\tau}}{\sum_{j \neq c} e^{z_j/\tau} + \beta e^{z_c/\tau}} \tag{4}$$

The parametric Softmax cross-entropy can be approximated as the following max function, as shown in (5). Minimizing this max function is expected that output z_c can be larger than other class outputs $z_j, j = 1, \ldots, m, j \neq c$, which is in line with the logic of the one-versus-all classification decision-making $cls\left(z\left(x\right)\right) = \max\left\{z_j\left(x\right)\right\}, j = 1, \ldots, m$.

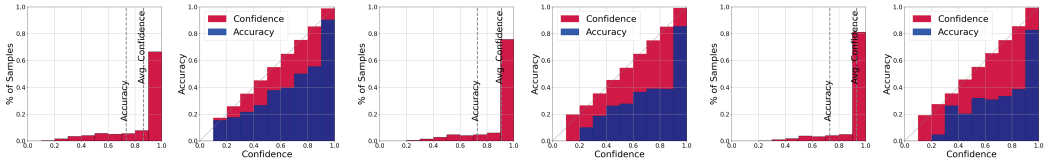
$$\lim_{\tau \to 0} -\log \frac{e^{z_c/\tau}}{\sum_{j \neq c} e^{z_j/\tau} + \beta e^{z_c/\tau}} = \lim_{\tau \to 0} \max\left\{\log \beta, z_j - z_c/\tau, j = 1, \ldots, m, j \neq c\right\} \tag{5}$$

The temperature coefficient τ has been extensively investigated in model calibration. Temperature scaling represents a commonly utilized calibration technique in post-processing calibration methods. β is regarded as a soft margin, with its approximation procedure detailed in Eq. (5). Consequently,

https://doi.org/10.52202/079017-3642



(a) $\beta = 20$ (b) $\beta = 10$ (c) $\beta = 5$



(d) $\beta = 1$ (e) $\beta = 0.1$ (f) $\beta = 0.01$

Figure 2: **Confidence and reliability diagrams with ResNet18 on CIFAR-100.** ($bins = 10$) In each subplot, the left plot illustrates the sample distribution in individual bins, while the right plot displays the average confidence and accuracy in each bin. Ideally, calibration aims for consistency between accuracy and average confidence in each bin. It indicates that a smaller gradient decay rate β is associated with more pronounced miscalibration of the model, while a larger gradient decay rate mitigates this issue.

CE can be interpreted as a margin-based loss function. Nonetheless, due to the distance distortion between input and representation spaces, maximizing the margin in the input space of models is not achieved simultaneously by large margin Softmax. Consequently, its dynamic characteristic in the optimization process tends to be ambiguous.

2.3 Probability-dependent gradient decay

Considering the Softmax with the sole hyperparameter β, the temperature τ is set to 1.

$$J = -\log \frac{e^{z_c}}{\sum_{j \neq c} e^{z_j} + \beta e^{z_c}} \tag{6}$$

Let us first consider the gradient of the Softmax.

$$\frac{\partial J}{\partial z_c} = -\frac{\sum e^{z_j} - e^{z_c}}{\sum e^{z_j} + (\beta - 1)\, e^{z_c}} \tag{7}$$

$$\frac{\partial J}{\partial z_j} = \frac{e^{z_j}}{\sum e^{z_j} + (\beta - 1)\, e^{z_c}} \tag{8}$$

Introducing probabilistic output $p_j = \frac{e^{z_j}}{e^{z_1} + \cdots + e^{z_m}}$ as an intermediate variable, we obtain

$$\frac{\partial J}{\partial z_j} = \begin{cases} -\dfrac{1 - p_c}{1 + (\beta - 1) p_c}, j = c \\ \dfrac{p_j}{1 + (\beta - 1) p_c}, j \neq c \end{cases} \tag{9}$$

Fig. 1 illustrates how the introduced hyperparameter β determines the gradient decay rate as the instance-level probability increases. More empirical experimental results can be found in the Appendix A.2. A smaller β results in a reduced decay rate of gradient amplitude corresponding to probability. Empirical investigations have shown that the magnitude of β during optimization determines the average confidence level and consequently impacts the calibration performance of the model. Fig. 2 demonstrates that a low gradient decay rate exacerbates the over-confidence in the model's output, whereas a high gradient decay rate can alleviate this issue and yield improved calibration results [27].

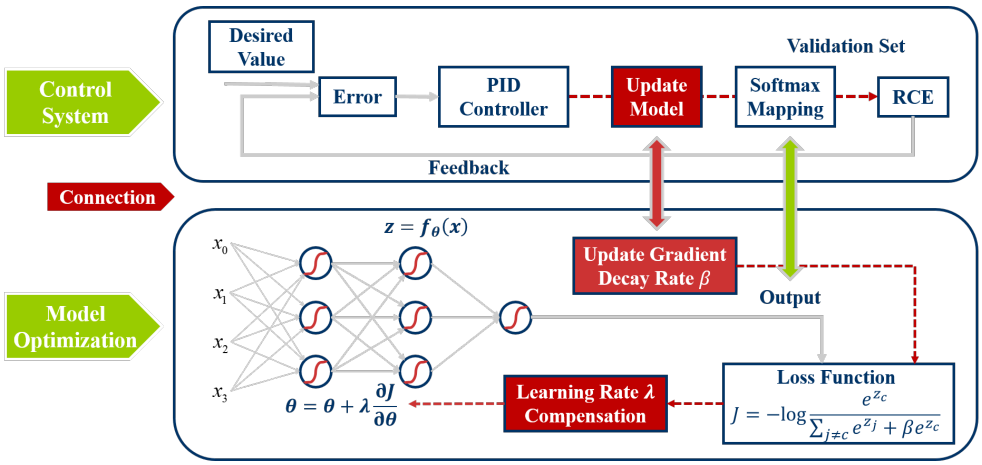


Figure 3: The framework of PID controller-based adaptive probability-dependent gradient decay.

3 Methodology

The probability-dependent gradient decay rate exhibits a negative correlation with the model's average confidence, as shown in Fig. 2 and Appendix A.2. Higher gradient decay rates result in a concave reduction in gradient magnitude as the sample confidence increases, thereby yielding a smoother distribution of confidence. Although high gradient decay rates can mitigate over-confidence distributions for samples, excessively small or large gradient magnitudes may lead to inadequate optimization. Consequently, we propose an adaptive gradient decay rate by a PID controller to ensure model calibration on optimization. Additionally, we propose a variable learning rate schedule to adjust the gradient and counterbalance the impact of fluctuating gradient decay rates on gradient magnitude. The whole framework is shown in Fig. 3.

3.1 PID controller approach for adaptive probability-dependent gradient decay

Our approach focuses on regulating the gradient decay rate by monitoring the average confidence and accuracy within each bin of the validation set throughout model optimization. This aims to enhance the model's calibration and dynamic properties during optimization. Eq. (1) specifies the desired calibration by representing only the absolute value of the difference between average confidence and accuracy within each bin. However, it fails to capture the under-confidence and over-confidence in model calibration. Therefore, we propose a Relative Calibration Error (RCE) to reflect over-confidence and under-confidence:

$$RCE = \sum_{b=1}^{M} \frac{|D_b|}{N} \left(conf\left(D_b\right) - acc\left(D_b\right) \right) \tag{10}$$

In the control system depicted in Fig. 3, RCE serves as the output, with the gradient decay coefficient acting as the controlled variable. The target RCE value is set to 0. During each iteration, the model processes the validation set to calculate the RCE. If the RCE is greater than 0, it indicates over-confidence in the probability distribution, necessitating an increase in the gradient decay rate during model optimization. Conversely, if the RCE is less than 0, it signifies under-confidence in the probability distribution, prompting a decrease in the gradient decay rate. The PID controller determines the specific adjustment required for the gradient decay coefficient β. A PID controller continually computes an error $e\left(t\right)$, representing the disparity between the desired optimal RCE and the control system output. It then applies a correction $u\left(t\right)$ to the system, incorporating proportional (P), integral (I), and derivative (D) terms of $e\left(t\right)$. Mathematically, there is:

$$u\left(t\right) = K_p e\left(t\right) + K_i \int_0^t e\left(t\right) dt + K_d \frac{\partial}{\partial t} e\left(t\right) \tag{11}$$

https://doi.org/10.52202/079017-3642

Algorithm 1: PID controller-based adaptive gradient decay

Data: Training set $\left\{x_i^{train}, y_i^{train}\right\}_{i=1}^{N}$; Validation set $\left\{x_i^{val}, y_i^{val}\right\}_{i=1}^{N_{val}}$; Classification model f_θ; Learning rate λ; Batch size N_B; Maximum iteration T_{max}; PID controller K_p, K_i and K_d.

Result: Classification model f_θ

Initialize θ, β, λ;
while $t < T_{max}$ **do**
  # Adjust gradient decay rate β by PID controller;
  RCE of validation set $\leftarrow$ Computing by Eq.(10);
  $u(t) \leftarrow$ Computing by Eq.(11);
  $\beta_t \leftarrow$ Computing by Eq.(12);
  # Adjust learning rate λ by gradient compensation;
  $\alpha(t) \leftarrow$ Computing by Eq.(14);
  $\lambda(t) \leftarrow$ Computing by Eq.(15);
  # Optimize neural network by gradient-descent;
  $\theta \leftarrow \theta + \lambda(t) \frac{\partial J}{\partial \theta}$;
end

where K_p, K_i and K_d are the gain coefficients on the P, I and D terms, respectively. The coefficients K_p, K_i and K_d determine the contributions of present, past and future errors to the current correction. $e(t) = -RCE(t)$ represents the error of the t th optimization epoch. Since $0 < \beta$, the updating step is described as follows:

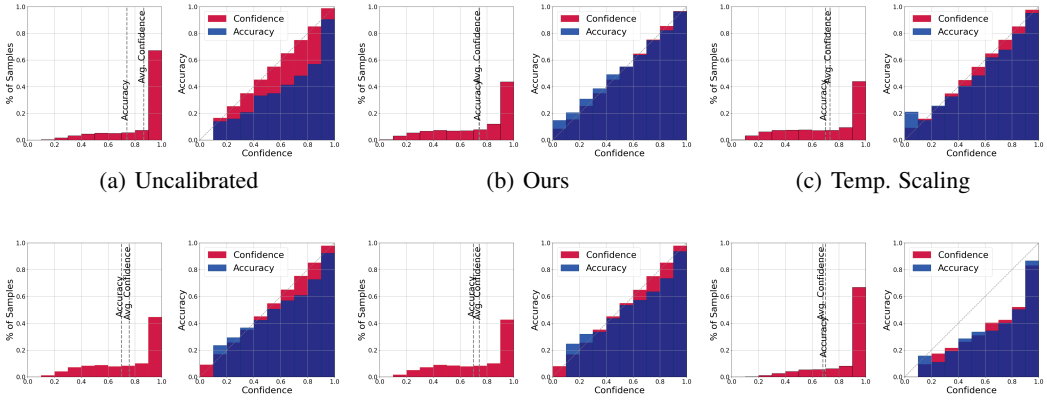
$$\beta_t = \beta_{t-1} e^{-u(t)} \tag{12}$$

The PID controller is a widely employed feedback control mechanism in engineering and industrial processes [30, 28]. It is utilized in systems requiring precise control over variables. As depicted in Fig. 3, the PID controller is utilized in the model optimization to regulate RCE, ensuring balanced model calibration and mitigating overfitting compared to model accuracy within the baseline strategy. As depicted in (11), the PID comprises three terms: the proportional (P) term, integral (I) term, and derivative (D) term. Firstly, the proportional (P) action ensures that the controller responds proportionally to the current error. It provides an immediate correction to minimize the RCE. Secondly, the integral (I) action continuously integrates the error over time and adjusts the control signal accordingly, eliminating any steady-state error. It eliminates any steady-state error by gradually reducing the cumulative error to zero. Thirdly, the derivative (D) action anticipates the future behavior of the error by considering its rate of change, damping oscillations and overshoots to improve system stability and transient response. PID controllers play a crucial role in maintaining stability, accuracy, and efficiency in ensuring precise model calibration by adjusting probability-dependent gradient decay in model optimization.

3.2 Adaptive learning rate for gradient compensation

Changes in the probability-dependent gradient decay rate significantly impact model calibration because varying decay rates change dynamic characteristic on the optimization for different samples. While a large gradient decay rate can maintain a similar confidence level in optimizing both high-confidence and low-confidence samples, a small decay rate results in a curriculum learning sequence where confidence in low-confidence samples increases only when confidence in high-confidence samples reaches a certain threshold. Nonetheless, varying decay rates also lead to changing gradient magnitudes, thereby influencing model optimization. For instance, a small gradient at the outset of optimization could result in insufficient optimization, thereby diminishing the model's generalization. To mitigate the influence of gradient magnitude fluctuations on model optimization, we implement a dynamic learning rate to counteract the impact of gradient fluctuations caused by varying β.

Since $\left|\frac{\partial J}{\partial z_c}\right| + \sum_{j \neq c} \left|\frac{\partial J}{\partial z_j}\right| = 2\left|\frac{\partial J}{\partial z_c}\right|$, $\left|\frac{\partial J}{\partial z_c}\right|$ can represent the gradient magnitude of the sample in output layer. Building on (9), we establish a metric to quantify the magnitude of the gradient at that specific gradient decay rate β:



(a) Uncalibrated (b) Ours (c) Temp. Scaling

(d) Matrix Scaling (e) Vector Scaling (f) Hist. Bin.

Figure 4: **Confidence histograms and reliability diagrams for different calibration methods with ResNet35 on CIFAR-100.** In each subplot, the left plot illustrates the sample distribution in individual bins, while the right plot displays the average confidence and accuracy in each bin. Our training calibration can improve performance on confidence estimate.

$$\alpha(t) = \int_0^1 \frac{1 - p_c}{1 + (\beta_t - 1)\, p_c} dp_c \tag{13}$$

Since $\beta > 0$, we obtain

$$\alpha(t) = \begin{cases} \frac{\beta_t \ln \beta_t - \beta_t + 1}{(\beta_t - 1)^2} & \beta_t \neq 1 \\ 0.5 & \beta_t = 1 \end{cases} \tag{14}$$

The learning rate, adjusted to account for the gradient dynamics over t epoch, is modified according to the following formula:

$$\lambda(t) = \lambda(t-1)\alpha(t-1)/\alpha(t) \tag{15}$$

where $\lambda(t)$ is learning rate of t th optimization epoch. The assumption underlying the gradient compensation proposed in (13)-(15) relies on the uniform distribution of samples within the probability interval $[0, 1]$. Integrating $\left|\frac{\partial J}{\partial z_c}\right|$ within the probability range $[0, 1]$ approximates the variation in gradient amplitudes across different gradient decay coefficients β. However, this assumption may not hold true during the optimization process. Nonetheless, this simple adaptive learning rate approach can effectively mitigate the issue of excessively small gradients resulting from overly large gradient decay rates. When the magnitude decreases, learning rate increases accordingly to compensate for the change in gradient amplitude. The whole Adaptive Probability-dependent Gradient Decay method by PID controller (PID-AGD) for model calibration is described in Algorithm 1.

4 Empirical experiments

The experimental validation comprises four main components. Firstly, we assess the performance of our algorithm and other calibration methods in terms of model calibration. Secondly, we compare the accuracy and model calibration results of our method with those obtained using different loss functions in supervised learning. Thirdly, we conduct ablation experiments on the proportional term K_p, integral term K_i, and differential term K_d across PID controller. Finally, we conducted the ablation experiment with different optimizers to verify the effectiveness of the adaptive learning rate for gradient compensation.

Train setting The baseline models include ResNet and VGG variants. The datasets comprised SVHN, CIFAR-10/100, 102 Flower and Tiny-ImageNet. In CIFAR-10/100, the training set contained 40,000 images, with testing and validation sets comprising 10,000 images each. The ratio of training set, validation set and test set in 102 Flower is 2:1:1 respectively. For Tiny-ImageNet, 100,000 images were used for training, and 10,000 images for testing and validation. SVHN utilized 58,606 training

https://doi.org/10.52202/079017-3642

Table 1: The calibration performance of different post-hoc calibration methods. The best results are in bold, and relative improvements over 2^{nd} best result in each section are in red. Results are averaged over five runs with different seeds.

Dataset	Model	Metric	Uncalibrated	Hist. Binning	Temp. Scaling	Vector Scaling	TS-AvUC	Ours
CIFAR-100	ResNet18	ECE	0.160±0.025	0.025±0.006	0.033±0.006	0.061±0.012	0.028±0.004	**0.006**(↓ ***0.019***)
		MCE	0.344±0.055	0.078±0.012	0.059 ±0.011	0.138 ±0.022	**0.052**(↓ ***0.007***)	0.068↑±0.018
		AdaECE	0.160±0.023	-	0.030±0.007	0.061±0.011	0.027±0.006	**0.007**(↓ ***0.020***)
CIFAR-100	ResNet35	ECE	0.172±0.027	0.034±0.009	0.026±0.004	0.056±0.011	0.035±0.008	**0.011** (↓ ***0.015***)
		MCE	0.351±0.061	0.064±0.010	**0.053**(↓ ***0.010***)	0.117±0.019	0.146±0.025	0.063↑±0.011
		AdaECE	0.172±0.028	-	0.027±0.006	0.053±0.010	0.034±0.007	**0.014**(↓ ***0.013***)
CIFAR-100	ResNet50	ECE	0.186±0.031	0.025±0.004	0.030±0.013	0.073±0.021	0.052±0.012	**0.016**(↓ ***0.009***)
		MCE	0.407±0.101	0.110±0.015	0.091±0.022	0.153±0.036	0.116±0.021	**0.056**(↓ ***0.035***)
		AdaECE	0.186±0.029	-	0.029±0.012	0.071±0.028	0.052±0.010	**0.006**(↓ ***0.023***)
CIFAR-100	VGG16	ECE	0.240±0.106	0.035±0.002	0.029±0.003	0.035±0.006	0.044±0.008	**0.013**(↓ ***0.016***)
		MCE	0.508±0.151	**0.042**(↓ ***0.002***)	0.093±0.029	0.084±0.009	0.101±0.026	0.044↑±0.003
		AdaECE	0.240±0.106	-	0.029±0.004	0.035±0.006	0.044±0.008	**0.012**(↓ ***0.017***)
CIFAR-10	ResNet35	ECE	0.054±0.010	0.011±0.001	0.015±0.002	0.014±0.003	0.015±0.006	**0.009**(↓ ***0.002***)
		MCE	0.300±0.085	0.255±0.102	0.121±0.026	**0.077**(↓ ***0.030***)	0.121±0.021	0.089↑±0.012
		AdaECE	0.054±0.011	-	0.014±0.004	0.013±0.002	0.013±0.005	**0.010**(↓ ***0.003***)
SVHN	ResNet18	ECE	0.021±0.006	0.016±0.002	0.009±0.003	0.007±0.002	0.010±0.003	**0.005**(↓ ***0.002***)
		MCE	0.286±0.053	**0.251**(↓ ***0.013***)	0.313±0.052	0.313±0.069	0.315±0.080	0.264↑±0.084
		AdaECE	0.021±0.006	-	0.010±0.003	0.009±0.002	0.013±0.005	**0.006**(↓ ***0.003***)
102 Flower	ResNet50	ECE	0.101±0.018	0.084±0.012	0.086±0.011	0.093±0.015	0.075±0.009	**0.033**(↓ ***0.042***)
		MCE	0.231±0.048	0.365±0.066	0.180±0.041	0.163±0.043	0.165±0.044	**0.132**(↓ ***0.031***)
		AdaECE	0.100±0.017	-	0.089±0.012	0.098±0.019	0.079±0.011	**0.031**(↓ ***0.048***)
Tiny-ImageNet	ResNet35	ECE	0.144±0.022	0.033±0.005	0.017±0.003	0.053±0.007	0.017±0.003	**0.009**(↓ ***0.008***)
		MCE	0.236±0.052	0.055±0.016	0.035±0.010	0.093±0.021	**0.030**(↓ ***0.005***)	0.035↑±0.006
		AdaECE	0.143±0.021	-	0.017±0.004	0.054±0.008	0.016±0.002	**0.010**(↓ ***0.014***)
VisDrone	YOLOv3	ECE	0.112 ±0.011	0.079 ±0.008	0.082 ±0.007	0.091±0.015	0.074 ±0.010	**0.044**(↓ ***0.030***)
		MCE	0.232 ±0.033	0.325 ±0.032	0.178 ±0.025	0.148 ±0.019	0.159 ±0.023	**0.136**(↓ ***0.012***)
		AdaECE	0.113 ±0.012	-	0.085 ±0.008	0.092 ±0.017	0.078 ±0.011	**0.046**(↓ ***0.032***)
COCO	YOLOv3	ECE	0.115 ±0.019	0.095 ±0.011	0.091 ±0.013	0.104 ±0.016	0.092 ±0.015	**0.077**(↓ ***0.014***)
		MCE	0.224 ±0.025	0.158 ±0.025	0.158 ±0.027	0.165 ±0.028	**0.164**(↓ ***0.001***)	0.174 ±0.030
		AdaECE	0.117 ±0.018	-	0.093 ±0.014	0.105 ±0.018	0.094 ±0.018	**0.078**(↓ ***0.015***)

images, 14,651 validation images, and 26,032 testing images. VisDrone contains 5471 training images, 1548 validation images and 3190 testing images. COCO utilized 82,783 training, 40,504 validation, and 40,775 testing images. In all classification experiments, the learning rate, momentum and weight clipping were set to 0.1, 0.9 and Norm=3, respectively. The learning rate decreased to 10% at 40% and 80% of the iterations, with weight decay set to 10^{-4} and a total of 200 iterations. For Tiny-ImageNet, the learning rate was set to 0.01 with a batch size of 64. The number of bins in all calibration metric are set to 10. P, I and D in our method are set to 1, 0.1, 1.

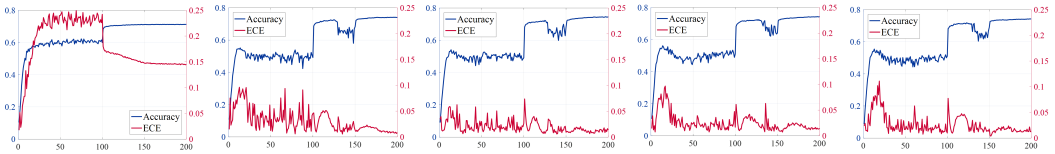
4.1 Calibration performance with other calibration methods

The experiments in this subsection aim to validate the efficacy of our proposed calibration method against baseline calibration methods, which employs PID control of the gradient decay rate. We compare our approach with other post-processing calibration methods, including Histogram Binning [31], Temperature Scaling [10], Vector Scaling, and TS-AvUC [13]. The evaluation metrics employed include ECE [29], MCE, and Adaptive Expected Calibration Error (AdaECE) [12]. The datasets primarily consist of CIFAR-10/100, SVHN, 102 Flower, Tiny-ImageNet, VisDrone and COCO, while the models predominantly belong to the VGG, ResNet and YOLO series. The comprehensive experimental results are presented in Table 1. The results of the visualization of all methods in ResNet35 on data CIFAR-100 for confidence histograms and reliability diagrams are presented in Fig. 4.

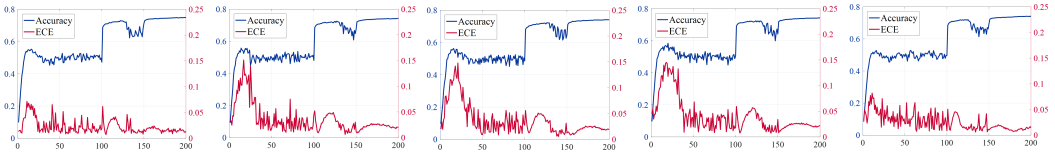
The results presented in Table 1 demonstrate that all methods effectively enhance the calibration of the model. However, post-processing calibration methods rely on an optimized independent output-probability mapping, which doesn't alter the optimization process of the original model itself. Consequently, these methods can solely refine the probability distribution of the model output. Our proposed method surpasses other calibration techniques in terms of overall ECE, MCE, and AdaECE. Therefore, these experimental findings underscore the effectiveness of our approach in enhancing model calibration by dynamically adjusting the gradient decay rate during the model optimization.

Table 2: The calibration performance and accuracy of different objective functions. The best results are in bold. Results are averaged over five runs with different seeds.

Methods	Models	CIFAR-10				CIFAR-100			
		ACC (%)	ECE	MCE	AdaECE	ACC (%)	ECE	MCE	AdaECE
Softmax	ResNet18	93.7±0.39	0.041±0.010	0.281±0.076	0.042±0.013	73.6±0.29	0.160±0.026	0.344±0.048	0.160±0.026
	ResNet35	93.9±0.39	0.054±0.015	0.300±0.083	0.054±0.016	73.8±0.30	0.172±0.022	0.351±0.077	0.172±0.023
	VGG16	92.1±0.41	0.066±0.022	0.339±0.091	0.068±0.023	69.2±0.26	0.233±0.054	0.476±0.112	0.236±0.053
Cosface	ResNet18	93.9±0.45	0.053±0.011	0.352±0.072	0.055±0.013	74.2±0.51	0.185±0.046	0.501±0.162	0.183±0.050
	ResNet35	**95.6**±0.42	0.048±0.012	0.317±0.095	0.049±0.011	74.6±0.38	0.181±0.065	0.488±0.127	0.178±0.063
	VGG16	92.7±0.58	0.067±0.019	0.390±0.101	0.068±0.020	71.4±0.52	0.238±0.081	0.567±0.125	0.233±0.085
Center loss	ResNet18	94.5±0.41	0.038±0.009	0.337±0.075	0.040±0.008	74.1±0.30	0.082±0.013	0.222±0.071	0.085±0.015
	ResNet35	95.5±0.51	0.043±0.010	0.280±0.099	0.045±0.012	74.2±0.31	0.098±0.031	0.250±0.096	0.101±0.030
	VGG16	**93.1**±0.41	0.034±0.009	0.349±0.083	0.034±0.010	**72.1**±0.37	0.216±0.042	0.472±0.104	0.231±0.045
DCA	ResNet18	91.9±0.32	0.020±0.006	0.156±0.038	0.022±0.007	72.1±0.25	0.047±0.011	0.156±0.024	0.049±0.012
	ResNet35	92.3±0.43	0.035±0.012	0.186±0.046	0.034±0.010	73.1±0.28	0.067±0.021	0.184±0.051	0.066±0.023
	VGG16	90.7±0.28	0.027±0.008	0.255±0.078	0.027±0.008	70.9±0.37	0.133±0.028	0.269±0.059	0.141±0.032
Ours	ResNet18	**95.0**±0.41	**0.007**±0.002	**0.078**±0.021	**0.008**±0.001	**74.3**±0.43	**0.006**±0.002	**0.068**±0.018	**0.007**±0.002
	ResNet35	95.6±0.51	**0.009**±0.002	**0.089**±0.012	**0.010**±0.003	**75.4**±0.39	**0.011**±0.003	**0.063**±0.011	**0.014**±0.002
	VGG16	92.6±0.35	**0.011**±0.002	**0.083**±0.031	**0.012**±0.004	71.9±0.35	**0.028**±0.008	**0.044**±0.003	**0.030**±0.010



(a) P:0 I:0 D:0 (b) P:1 I:1 D:1 (c) P:1 I:2 D:1 (d) P:1 I:0.1 D:1 (e) P:1 I:0.01 D:1



(f) P:1 I:0.01 D:2 (g) P:1 I:0.5 D:0.1 (h) P:1 I:0.5 D:0.01 (i) P:0.5 I:0.5 D:0.1 (j) P:5 I:0.5 D:0.1

Figure 5: **Accuracy and ECE of different PID settings with ResNet35 on CIFAR-100.** The preceding figures illustrate the testing accuracy and ECE outcomes in the model optimization. Notably, the accuracy appears insensitive to the PID controller configuration. Nonetheless, excessive settings of P, I, and D may compromise the stability of ECE during the model optimization.

4.2 Consistent optimization in supervised learning

Model accuracy and calibration are typically considered as two independent optimization metrics. In the baseline optimization strategy employing cross-entropy as the loss function, the model calibration tend to overfit earlier than the accuracy. To examine the impact of different optimization strategies on model calibration characteristics, Table 2 illustrates these effects. Comparative algorithms such as cosface [32], center loss [33], CE with DCA [34] and Softmax-based cross-entropy were utilized, and their performance in accuracy and uncertainty estimation was observed. While cosface, center loss, and our proposed approach, showed improvements in accuracy, the former two algorithms did not consider model calibration, resulting in overconfident predictions. DCA regularity improves model calibration but hurts accuracy. Conversely, our proposed PID-based method with variable gradient decay rate ensures both model accuracy and calibration. This reaffirms the significant influence of probability-dependent gradient decay rates on model calibration and overall performance.

4.3 Ablation experiments and analysis for PID controller

In the model optimization, both model optimization and gradient coefficient decay need consideration, constituting a bi-level optimization problem. From a control system perspective, altering the gradient decay rate modifies the dynamic characteristics of subsequent iterations in the model optimization. However, its impact on model calibration may not be fully apparent in the immediate optimization

https://doi.org/10.52202/079017-3642

Table 3: Different optimizer performance in ResNet35 on CIFAR-100. Results are averaged over five runs with different seeds. Adam optimization compromises model accuracy when applied to a dynamic optimization objective using a PID controller approach.

SGD	Adam	PID Controller Approach	Gradient Compensation	Accuracy	ECE	AdaECE
✓	-	-	-	73.8%	0.172	0.172
✓	-	✓	-	72.5%	0.022	0.023
-	✓	✓	-	63.5%	0.023	0.024
✓	-	✓	✓	74.7%	0.012	0.013

results and may require more iterations for observation. Thus, viewed from this perspective, the entire control system can be regarded as having a time lag, whereby the controlled variable β exhibits a certain delay concerning the RCE. While it may be challenging to mathematically describe the control system, the PID controller serves as a "black-box" controller, leveraging the integral and differential variations of the error, proving highly effective. The ablation experiments are shown in Fig. 5. We conclude that model calibration is robust to the choice of PID parameters; however, setting the PID parameters too high can compromise the stability of ECE during model optimization. Based on this, we selected PID settings of 1, 0.1, and 1 for the experiments presented above. On the other hard, in Fig. 5, the accuracy curves with our method all exhibited a noticeable jitter, which requires further investigation.

4.4 Ablation experiments of different optimizer

The motivation for proposing the adaptive learning rate for gradient compensation arises from the observation that significant variations in gradient magnitude can negatively impact the model's optimization for classification error. A dynamic learning rate helps maintain a relatively stable gradient magnitude. Additionally, the Adam optimizer aids in reducing gradient fluctuations. To verify the novelty of our method, Table 3 presents the performance of various optimizers when applied to our proposed PID controller-based calibration method.

Our experimental results indicate that Adam can indeed provide a more stable gradient and calibration performance, particularly in conjunction with our PID controller approach for model calibration. However, it is notable that Adam results in reduced accuracy, achieving only 63.5% on CIFAR-100 with ResNet35, significantly lower than the baseline accuracy of 73.8%. A key difference arises in the baseline case handled by Adam. In our proposed PID controller method, which adjusts the hyperparameter β during model calibration, the loss function is dynamic. While Adam retains previous gradient information, this can conflict with the current gradient vector direction because the optimization objective is dynamic. In contrast, our compensation method only modifies the learning rate and retains gradient vector direction pertinent to the current loss function. This may explain why the Adam optimizer does not yield better results.

5 Conclusion

The gradient decay rate plays a crucial role in shaping the calibration characteristics and uncertainty distribution of deep learning throughout the dynamic optimization. Our results show a negative correlation between the gradient decay rate with increasing instance-level probability and the overall confidence distribution. This paper introduces a novel optimization approach aimed at regulating the gradient decay rate hyperparameter β, via a PID controller. The goal is to achieve perfect model calibration by monitoring the proposed relative calibration error of the validation set. Within this control system framework as shown in Fig. 3, the probabilistic gradient decay rate serves as the controlled variable, while a newly defined relative calibration error acts as the control system output, mitigating both over-confidence and under-confidence in the model. Additionally, to address fluctuations of gradient amplitude resulting from varying gradient decay rate, a new learning rate compensation mechanism is employed. Empirical validation demonstrates that our proposed adaptive gradient decay rate optimization strategy, facilitated by a PID controller, effectively enhances both the accuracy and model calibration in deep learning, ensuring adequate calibration throughout the supervised learning.

References

[1] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[2] Gustavo Carneiro, Leonardo Zorron Cheng Tao Pu, Rajvinder Singh, and Alastair Burt. Deep learning uncertainty and confidence calibration for the five-class polyp classification from colonoscopy. *Medical Image Analysis*, 62:101653, 2020.

[3] Jian Xu, Xinxiong Jiang, Siyang Liao, Deping Ke, Yuanzhang Sun, Liangzhong Yao, and Beilin Mao. Probabilistic prognosis of wind turbine faults with feature selection and confidence calibration. *IEEE Transactions on Sustainable Energy*, 15(1):52–67, 2024.

[4] Dimitrios Milios, Raffaello Camoriano, Pietro Michiardi, Lorenzo Rosasco, and Maurizio Filippone. Dirichlet-based gaussian processes for large-scale calibrated classification. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[5] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2019.

[6] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

[7] Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. In *Advances in Neural Information Processing Systems*, volume 34, pages 11809–11820, 2021.

[8] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial Intelligence and Statistics*, pages 623–631. PMLR, 2017.

[9] Yongqiao Wang, Lishuai Li, and Chuangyin Dang. Calibrating classification probabilities with shape-restricted polynomial regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1813–1827, 2019.

[10] Christian Tomani, Daniel Cremers, and Florian Buettner. Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In *European Conference on Computer Vision*, pages 555–569. Springer, 2022.

[11] Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. Intra order-preserving functions for calibration of multi-class neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 13456–13467, 2020.

[12] Khanh Nguyen and Brendan O'Connor. Posterior calibration and exploratory analysis for natural language processing models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1587–1598, 2015.

[13] Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 18237–18248, 2020.

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

[15] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019.

[16] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

 https://doi.org/10.52202/079017-3642

[17] Aditya Krishna Menon, Ankit Singh Rawat, Sashank J Reddi, Seungyeon Kim, and Sanjiv Kumar. Why distillation helps: a statistical perspective. *arXiv preprint arXiv:2005.10419*, 2020.

[18] Linwei Tao, Minjing Dong, Daochang Liu, Changming Sun, and Chang Xu. Calibrating a deep neural network with its predecessors. *arXiv preprint arXiv:2302.06245*, 2023.

[19] Bowen Lei, Ruqi Zhang, Dongkuan Xu, and Bani Mallick. Calibrating the rigged lottery: Making all tickets reliable. In *International Conference on Learning Representations*, 2022.

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

[21] Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *International Conference on Machine Learning*, pages 7034–7044. PMLR, 2020.

[22] Bingyuan Liu, Jérôme Rony, Adrian Galdran, Jose Dolz, and Ismail Ben Ayed. Class adaptive network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16070–16079, 2023.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2980–2988, 2017.

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

[25] Arindam Ghosh, Thomas Schaaf, and Matthew Gormley. Adafocal: Calibration-aware adaptive focal loss. In *Advances in Neural Information Processing Systems*, volume 35, pages 1583–1595, 2022.

[26] Jonathan Wenger, Hedvig Kjellström, and Rudolph Triebel. Non-parametric calibration for classification. In *International Conference on Artificial Intelligence and Statistics*, pages 178–190. PMLR, 2020.

[27] Siyuan Zhang and Linbo Xie. Advancing neural network calibration: The role of gradient decay in large-margin softmax optimization. *Neural Networks*, 178:106457, 2024.

[28] Wangpeng An, Haoqian Wang, Qingyun Sun, Jun Xu, Qionghai Dai, and Lei Zhang. A pid controller approach for stochastic optimization of deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8522–8531, 2018.

[29] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[30] Kiam Heong Ang, Gregory Chong, and Yun Li. Pid control system analysis, design, and technology. *IEEE transactions on control systems technology*, 13(4):559–576, 2005.

[31] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning*, volume 1, pages 609–616, 2001.

[32] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[33] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *Computer vision–ECCV 2016: 14th European conference*, pages 499–515, 2016.

[34] Gongbo Liang, Yu Zhang, Xiaoqin Wang, and Nathan Jacobs. Improved trainable calibration method for neural networks on medical imaging classification. In *British Machine Vision Conference 2020*.

A Appendix

A.1 Experiments compute resources

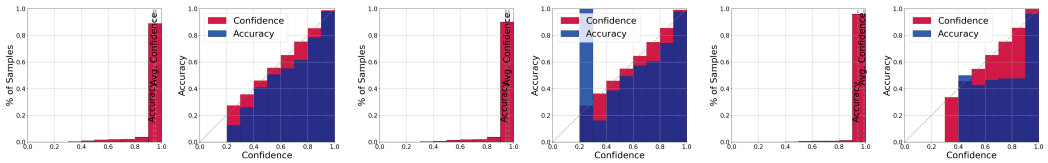
For experiments, we utilized compute resources featuring an NVIDIA A100 GPU with PCIe interface and 40GB memory capacity, accompanied by PyTorch version 1.7.0 with CUDA version 11.0. The computational backbone was supported by an Intel Xeon Gold 6278C processor.

A.2 Experiments for fixed gradient decay rate

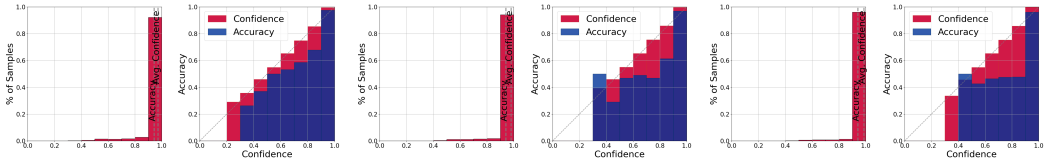
More empirical results validating the relationship between gradient decay hyperparameter β and model calibration are presented in Tab. 4, Tab. 5 and Tab. 6, while confidence histograms and reliability diagrams are depicted in Figs. 6-9. These experiments consistently conclude that the gradient decay rate β is negatively correlated with the overall confidence level of the model. Specifically, when the value β is small, the gradient decay rate decreases, leading to higher overall confidence in the model output and a greater likelihood of overconfident probabilistic output. Conversely, as the value β increases, the gradient decay rate increases, resulting in lower overall confidence in the model's output and a higher probability of underfitting in the model's probabilistic output distribution. Although model accuracy appears somewhat linked to this hyperparameter, conclusive generalizations from theoretical perspective cannot be drawn for the current experimental results.

Table 4: Model calibration of different gradient decay and post-processing calibration. The best results are in bold. Results are averaged over five runs with different seeds. ($bins = 10$)

Dataset	Model	Metric	Gradient decay factor β				Vector Scaling	Temp. Scaling
			20	10	1	0.1		
CIFAR-100	ResNet18	ECE	**0.019**±0.003	0.048±0.008	0.111±0.011	0.161±0.021	0.039±0.006	0.026±0.005
		MCE	**0.063**±0.011	0.139±0.025	0.306±0.051	0.423±0.076	0.135±0.036	0.064±0.021
CIFAR-100	ResNet34	ECE	**0.026**±0.004	0.055±0.008	0.131±0.019	0.182±0.022	0.042±0.006	0.038±0.005
		MCE	0.087±0.011	0.162±0.031	0.233±0.068	0.332±0.091	0.131±0.032	**0.059**±0.011
CIFAR-100	VGG16	ECE	0.122±0.009	0.163±0.011	0.207±0.031	0.226±0.033	0.030±0.008	**0.022**±0.005
		MCE	0.317±0.021	0.378±0.051	0.499±0.088	0.556±0.093	0.523±0.109	**0.041**±0.011
CIFAR-10	ResNet18	ECE	0.021±0.004	0.025±0.006	0.036±0.010	0.042±0.011	**0.011**±0.003	0.015±0.003
		MCE	0.591±0.153	0.268±0.095	0.295±0.068	0.355±0.111	**0.051**±0.012	0.089±0.009
Tiny-ImageNet	ResNet34	ECE	**0.014**±0.002	0.036±0.005	0.089±0.011	0.226±0.056	0.017±0.006	0.019±0.004
		MCE	**0.035**±0.005	0.069±0.009	0.166±0.021	0.382±0.079	0.036±0.010	0.063±0.013
Tiny-ImageNet	ResNet50	ECE	0.041±0.007	**0.013**±0.002	0.104±0.021	0.188±0.032	0.023±0.004	0.027±0.003
		MCE	0.082±0.011	0.044±0.009	0.149±0.020	0.377±0.045	**0.039**±0.011	0.067±0.016



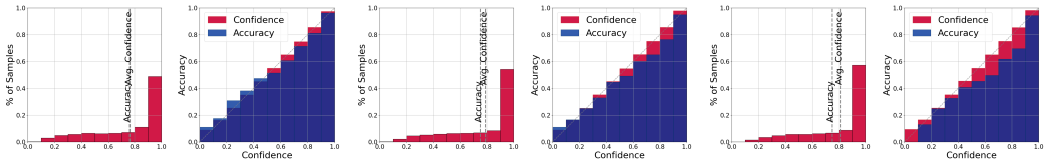
(a) $\beta = 50$ (b) $\beta = 20$ (c) $\beta = 10$



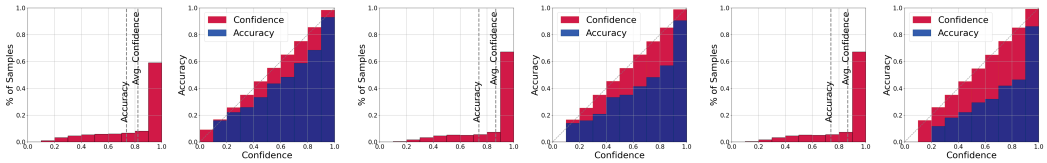
(d) $\beta = 5$ (e) $\beta = 1$ (f) $\beta = 0.1$

Figure 6: Confidence histograms and reliability diagrams for gradient decay with ResNet18 on CIFAR-10. ($bins = 10$)

https://doi.org/10.52202/079017-3642

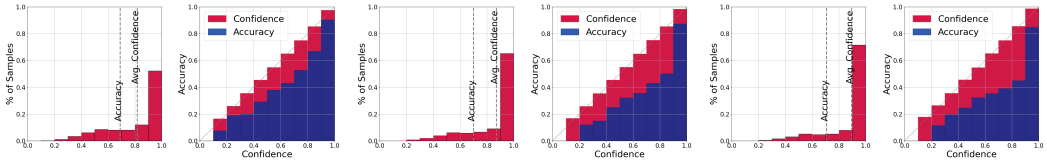


(a) $\beta = 50$ (b) $\beta = 20$ (c) $\beta = 10$

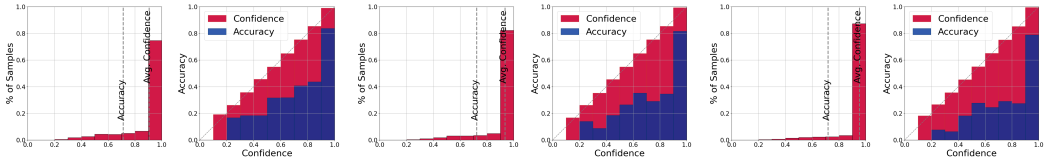


(d) $\beta = 5$ (e) $\beta = 1$ (f) $\beta = 0.1$

Figure 7: Confidence histograms and reliability diagrams with ResNet34 on CIFAR-100. ($bins = 10$)

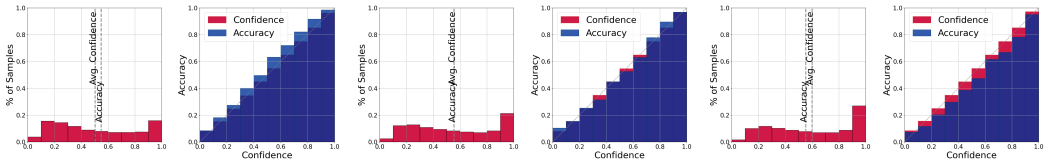


(a) $\beta = 50$ (b) $\beta = 20$ (c) $\beta = 10$

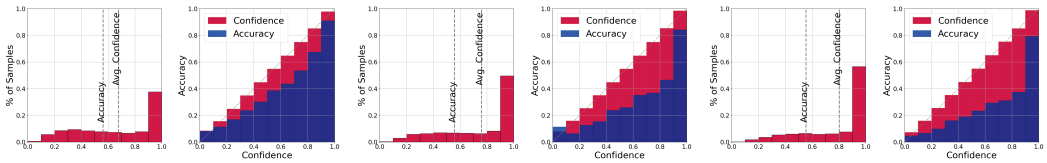


(d) $\beta = 5$ (e) $\beta = 1$ (f) $\beta = 0.1$

Figure 8: Confidence histograms and reliability diagrams for gradient decay with VGG16 on CIFAR-100. ($bins = 10$)



(a) $\beta = 20$ (b) $\beta = 10$ (c) $\beta = 5$



(d) $\beta = 1$ (e) $\beta = 0.1$ (f) $\beta = 0.01$

Figure 9: Confidence histograms and reliability diagrams with ResNet50 on Tiny-ImageNet. ($bins = 10$)

Table 5: The performance of ResNet34 on Tiny-ImageNet with different gradient decay. The best results are in bold. Results are averaged over five runs with different seeds. ($bins = 10$)

Metric	Gradient decay factor β					
	20	10	5	1	0.1	0.01
Top-1 Acc (%)	52.8	53.2	**53.8**	53.4	53.7	53.6
Top-5 Acc (%)	**75.8**	75.5	75.1	75.0	74.4	73.2
Training Acc (%)	88.6	88.5	89.7	90.3	**90.9**	90.5
ECE ($bins = 10$)	**0.015**	0.034	0.076	0.087	0.224	0.274
MCE ($bins = 10$)	**0.036**	0.065	0.151	0.176	0.406	0.518

Table 6: The performance of ResNet50 on Tiny-ImageNet with different gradient decay. The best results are in bold. Results are averaged over five runs with different seeds. ($bins = 10$)

Metric	Gradient decay factor β					
	20	10	5	1	0.1	0.01
Top-1 Acc (%)	55.9	56.0	**56.4**	56.3	56.4	56.2
Top-5 Acc (%)	77.7	**78.0**	77.3	76.6	76.0	74.9
Training Acc (%)	86.4	88.8	90.3	91.9	**92.3**	91.8
ECE ($bins = 10$)	0.045	**0.014**	0.046	0.114	0.203	0.249
MCE ($bins = 10$)	0.084	**0.044**	0.082	0.151	0.388	0.476

A.3 Limitations and future works

Our work currently lacks theoretical analysis. Although all experimental findings consistently demonstrate the impact of gradient decay rate β on model calibration, we still require theoretical frameworks to explain how the gradient decay rate affects the overall confidence distribution. Our experiments indicate that large gradient decay rates result in similar confidence levels across samples, while smaller rates yield more discriminatory levels. In essence, smaller decay rates enforce a more stringent curriculum learning sequence, whereby increased confidence in difficult samples only occurs after optimizing easier ones. Consequently, this leads to greater differentiation in final confidence distribution of different samples. While these empirical observations are compelling, they lack theoretical substantiation.

Methodologically, our current approach employs a PID controller to control gradient decay in optimization. However, in practice, the effect of gradient decay rate adjustments on model calibration requires a large number of epochs to manifest changes. Viewed from a control systems perspective, this delay indicates a substantial time lag in the control object. However, as neural network optimization processes defy mathematical description, designing effective controllers becomes inherently challenging. Furthermore, our work also lacks comprehensive statistical analysis of the calibrated outputs. Future research should address how alterations in the dynamic gradient decay rate impact the internal optimization process.

Additionally, the temperature coefficient τ also impacts model calibration. The Softmax with small τ disperses the inter-class distance by adjusting the probability output to focus more on hard negative samples. Nevertheless, large τ can only smooth the output of all categories and cannot mine more information from simple positive samples. On the contrary, small β makes the gradient decay slowly so that easy positive samples can be sufficiently learned up to high confidence. An appropriate β can mining more discriminative features on the whole. Similarly, large β only keeps the samples at the same level of confidence and cannot extract more meaningful features from challenging samples. τ and β improved the mining capability of Softmax in two different dimensions. The exploration of optimizing the effects of both hyperparameters to improve model calibration represents a promising approach.

https://doi.org/10.52202/079017-3642

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Motivation and two technical contributions are articulated in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please find limitations and future works in Appendx A.3.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please find Train setting in Section 4 and Algorithm 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

https://doi.org/10.52202/079017-3642

Answer: [Yes]

Justification: The code is available in https://github.com/UHIF/PID_AGD. Please find Abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please find Train setting in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The descriptions of the figures are in the caption. All metrics are defined, such as (1)-(2) or added relevant references.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please find Appendix A.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper is foundational research and does not deal with applications.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

https://doi.org/10.52202/079017-3642

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

 https://doi.org/10.52202/079017-3642