Class Adaptive Network Calibration

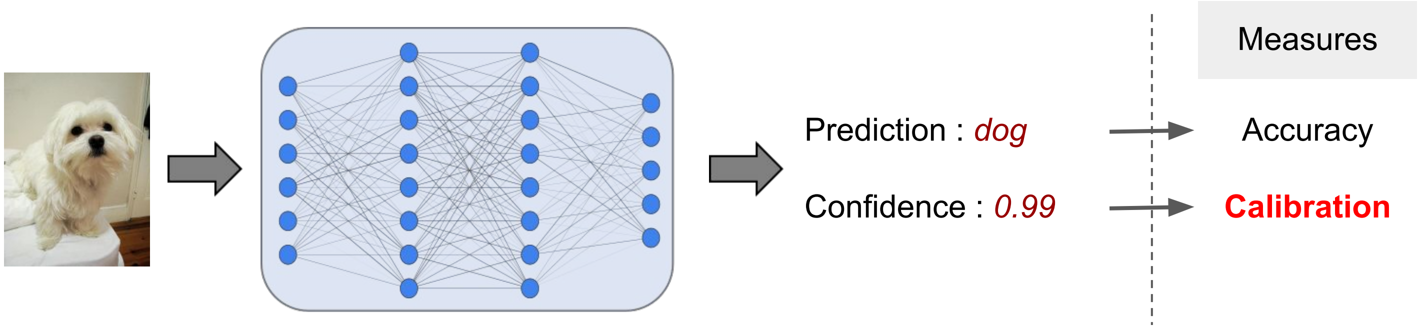
Bingyuan Liu[*1], Jérôme Rony[*1], Adrian Galdran[2], Jose Dolz[1], Ismail Ben Ayed[1]

[1]ÉTS Montreal, Canada [2]Universitat Pompeu Fabra, Spain

Poster info : WED-PM-354

Background: calibration

- Calibration should be considered beyond accuracy



Perfectly calibrated : Confidence = Accuracy

Miscalibrated:

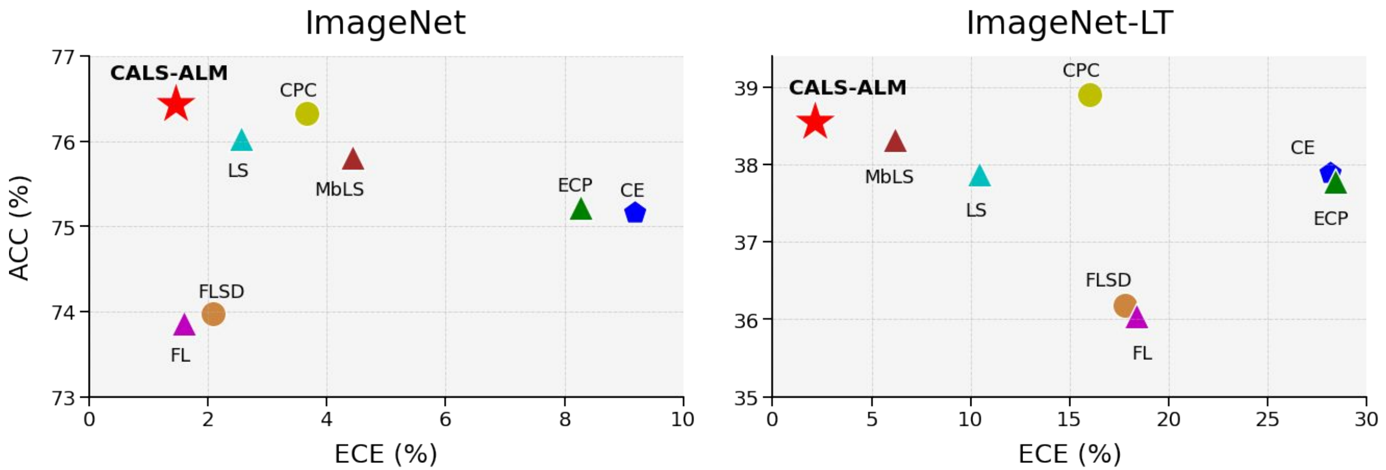
- Over-confident : Confidence > Accuracy
- Under-confident : Confidence < Accuracy

Background: calibration

- Previous methods: Focus on enhancing learning objectives or employing post-processing technique

- Our message: we established that a better designed optimizer notably impacts the outcomes.

Contributions

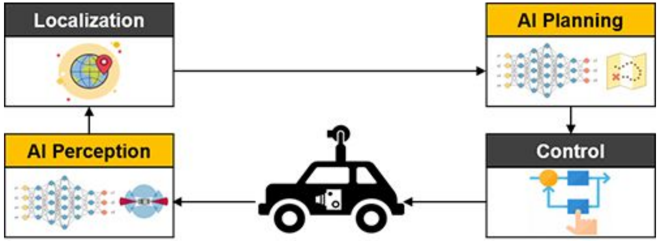


- CALS: Propose Class Adaptive Label Smoothing, where adaptive class-wise multipliers are introduced.
- ALM: Implement a modified Augmented Lagrangian Multiplier algorithm to solve the constrained optimization problem of calibration.
- SOTA: Superior calibration performances over a variety of benchmarks.

Introduction - calibration matters

- Deep learning models as a service
  - providing reliable prediction confidence for customers or downstream modules

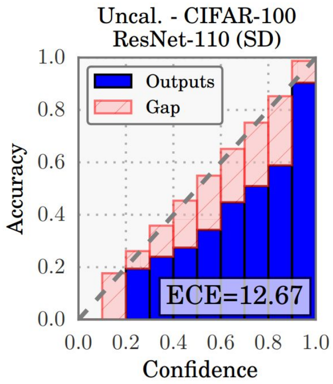
- Safety-sensitive applications :



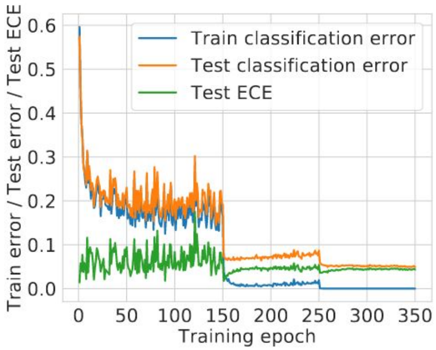
Autonomous driving

Medical diagnosis

DNNs are poorly calibrated !



[Guo et al., ICML 2017]



[Mukhoti et al., NeurIPS 2020]

Causes of Miscalibration :

- Cross-entropy objective : push the predictions to match the binary ground-truth
- Over-fitting of high-capacity DNNs in probabilistic error rather than classification error

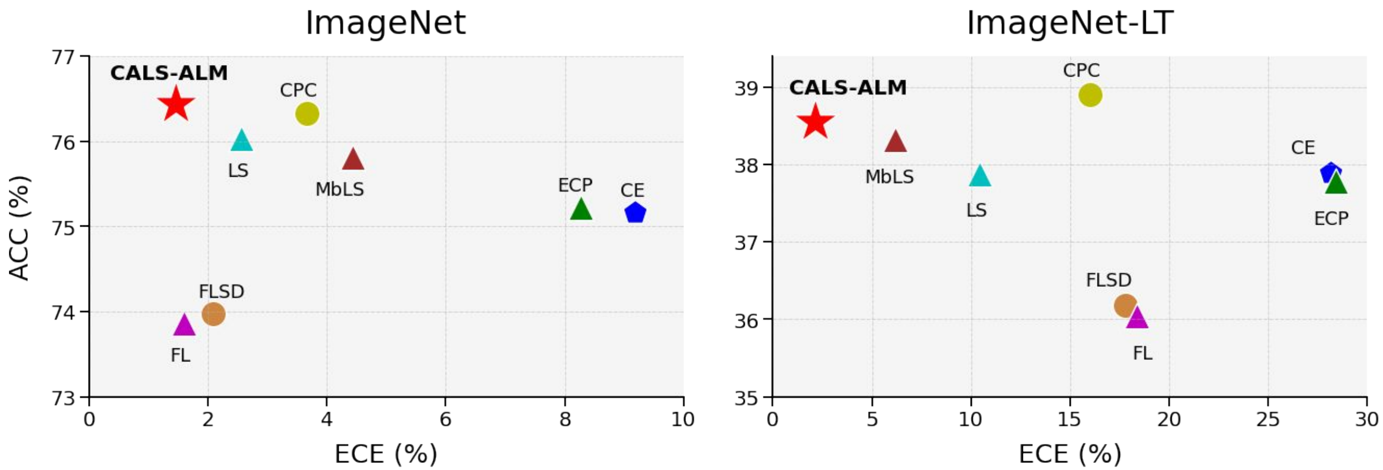
Previous solutions

- Post-processing method :
    - Temperature scaling [Guo et al., ICML 2017]
    - Post-hoc uncertainty calibration for domain drift scenarios [Tomani et al., CVPR 2021]
    - Local temperature scaling [Ding et al., ICCV 2021]
- Training methods:
    - Explicitly penalizes the prediction by maximizing its entropy (ECP) [Pereyra et al., ICLR 2017]
    - Label smoothing (LS) [Muller et al., NeurIPS 2019]
    - Focal loss (FL) [Mukhoti et al., NeurIPS 2020]
    - CPC [Cheng et al., CVPR 2022]
    - MbLS [Liu et al., CVPR 2022]

Our motivation

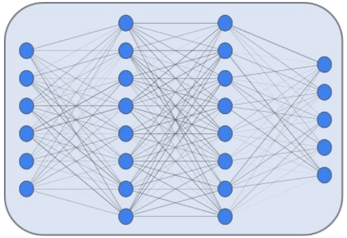
- Instead of exploring a better learning objective, we establish that the optimizer matters significantly
- The scalar balancing weight used for controlling relative contribution of calibration penalty is not ideal:
  - The weight is the same for all classes
  - The weight is usually fixed without an adaptive strategy

Contributions

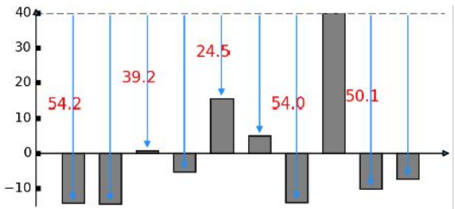


- CALS: Propose Class Adaptive Label Smoothing, where adaptive class-wise multipliers are introduced.
- ALM: Implement a modified Augmented Lagrangian Multiplier algorithm to solve the constrained optimization problem of calibration.
- SOTA: Superior calibration performances over a variety of benchmarks.

Notations



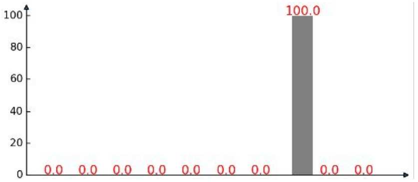
DNN



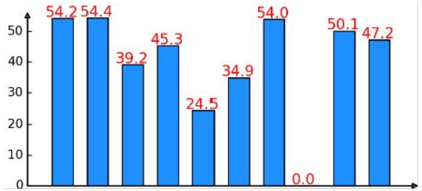
logit : $\mathbf{l} = (l_k)_{1 \leq k \leq K} \in \mathbb{R}^K$

Softmax

$$s_k = \frac{\exp^{l_k}}{\sum_j^K \exp^{l_j}}$$



prediction : $\mathbf{s} = (s_k)_{1 \leq k \leq K} \in \mathbb{R}^K$



logit distances

$$\mathbf{d}(\mathbf{l}) = (\max_j(l_j) - l_k)_{1 \leq k \leq K} \in \mathbb{R}^K$$

Cross Entropy

$$\mathcal{L}_{\mathrm{CE}} = -\sum_k y_k \log s_k$$

Formulation

The constrained optimization for training calibrated network:

$$\begin{aligned} \min_{\theta} \quad & \sum_{i=1}^{N} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{x}^{(i)}, y^{(i)}) \\ \mathrm{s.\,t.} \quad & \max_{k}\{\boldsymbol{l}_k^{(i)}\} - l^{(i)} \preceq m\mathbf{1}_K, \quad i = 1, \ldots, N, \end{aligned}$$

Previous metod (MbLS [Liu et al. CVPR 2022]) approximately solves it by a penalty method.

We propose to address it by Augmented Lagrangian Multiplier (ALM).

Formulation

General constrained optimization:

$$\min_{x} \quad f(x) \quad \text{s.t.} \quad h_i(x) \le 0, \quad i = 1, \dots, n$$

General ALM:

$$\min_{x} \quad \mathcal{L}^{(j)}(x) = f(x) + \sum_{i=1}^{n} P(h_i(x), \rho_i^{(j)}, \lambda_i^{(j)})$$

Algorithm 1 Augmented Lagrangian Multiplier algorithm

Require: Objective function f
Require: Constraint functions $h_i, i = 1, \dots, n$
Require: Penalty function P, initial $\boldsymbol{\lambda}^{(0)} \in \mathbb{R}^n_{++}$, $\boldsymbol{\rho}^{(0)} \in \mathbb{R}^n_{++}$
Require: Initial variable $x^{(0)}$, iterations $j = 1$

while not converged **do**
  Initialize with $x^{(j-1)}$ and minimize (approximately):
  $\mathcal{L}^{(j)}(x) = f(x) + \sum_{i=1}^{n} P(h_i(x), \rho_i^{(j)}, \lambda_i^{(j)})$
  $x^{(j)} \leftarrow$ (approximate) minimizer of $\mathcal{L}^{(j)}$
  for $i = 1, \dots, n$ **do**
    $\lambda_i^{(j+1)} \leftarrow P'(h_i(x^{(j)}), \rho_i^{(j)}, \lambda_i^{(j)})$
    if the i-th constraint does not improve **then**
      $\rho_i^{(j+1)} \leftarrow \gamma \rho_i^{(j)}$
    else
      $\rho_i^{(j+1)} \leftarrow \rho_i^{(j)}$
    end if
  end for
  $j \leftarrow j + 1$
end while

Formulation

The critical designs to make ALM applicable for training DNNs:

- Apply class-wise multipliers instead of sample-wise multipliers.
- Consider that a training epoch corresponds to the approximate minimization of the loss function.
- Use the validation set to obtain a reliable estimate of the penalty multipliers at each epoch.
- Utilize the PHR function as the penalty function.

Formulation

The formulation of the proposed CALS-ALM:

$$\sum_{i=1}^{N} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{x}^{(i)}, y^{(i)}) + \frac{1}{K} \sum_{k=1}^{K} P\left(\frac{d_k^{(i)}}{m} - 1, \rho_k, \lambda_k\right)$$

PHR penalty function:

$$\mathrm{PHR}(z, \rho, \lambda) = \begin{cases} \lambda z + \frac{1}{2}\rho z^2 & \text{if} \quad \lambda + \rho z \geq 0; \\ -\frac{\lambda^2}{2\rho} & \text{otherwise.} \end{cases}$$

Experiments

Datasets:

- Image classification: Tiny-ImageNet, ImageNet
- Long-tailed image classification: ImageNet-LT
- Semantic segmentation: PASCAL VOC 2012
- Text classification: 20 Newsgroups

Metrics:

- Calibration: expected calibration error (ECE) and its variant, Adaptive ECE (AECE)
- Discrimination:
  - Accuracy (ACC) for classification
  - Mean interaction over union (mIoU) for segmentation

Experiments

- Best calibration performance with almost no cost in accuracy

	TinyImageNet			ImageNet						ImageNet-LT					
	ResNet-50			ResNet-50			SwinV2-T			ResNet-50			SwinV2-T		
Method	Acc	ECE	AECE	Acc	ECE	AECE	Acc	ECE	AECE	Acc	ECE	AECE	Acc	ECE	AECE
CE	65.02	3.73	3.69	75.16	9.19	9.18	75.60	9.95	9.94	37.90	28.12	28.12	31.82	31.82	36.68
MMCE [18]	<u>65.34</u>	2.81	2.61	74.85	8.57	8.56	76.68	9.07	9.08	37.79	28.41	28.40	33.14	26.41	26.41
ECP [38]	64.90	4.00	3.92	75.22	8.27	8.26	75.82	9.88	9.86	37.69	28.14	28.13	31.22	33.70	33.70
LS [42]	**65.78**	3.17	3.16	76.04	2.57	2.88	75.42	7.32	7.33	37.88	10.46	10.38	31.70	11.42	11.40
FL [31]	63.09	2.96	3.12	73.87	<u>1.60</u>	<u>1.65</u>	75.60	3.19	3.18	36.04	18.37	18.36	30.73	25.50	25.50
FLSD [31]	64.09	2.91	2.95	73.97	2.08	2.06	74.70	2.44	2.37	36.18	17.77	17.78	32.56	25.16	25.17
CPC [5]	64.49	4.88	4.91	76.33	3.66	3.59	76.34	5.50	5.33	**38.90**	16.00	15.99	32.54	13.21	13.19
MbLS [22]	64.74	<u>1.64</u>	<u>1.73</u>	75.82	4.44	4.26	<u>77.18</u>	<u>1.95</u>	<u>1.73</u>	38.32	6.16	6.16	32.05	7.65	7.64
CALS-HR	65.09	2.50	2.42	<u>76.34</u>	5.63	5.69	**77.58**	3.06	2.95	38.50	<u>2.83</u>	<u>2.78</u>	**34.31**	<u>2.37</u>	**2.45**
CALS-ALM	65.03	**1.54**	**1.38**	**76.44**	**1.46**	**1.32**	77.10	**1.61**	**1.69**	<u>38.56</u>	**2.15**	**2.30**	<u>33.94</u>	**2.32**	**2.45**

Table 1. Calibration performance for different approaches on three image classification benchmarks. We report two lower-is-better calibration metrics, *i.e*. ECE and AECE. Best method is highlighted in bold, while the second-best one is underlined.

Experiments

- Visualization of learned multipliers

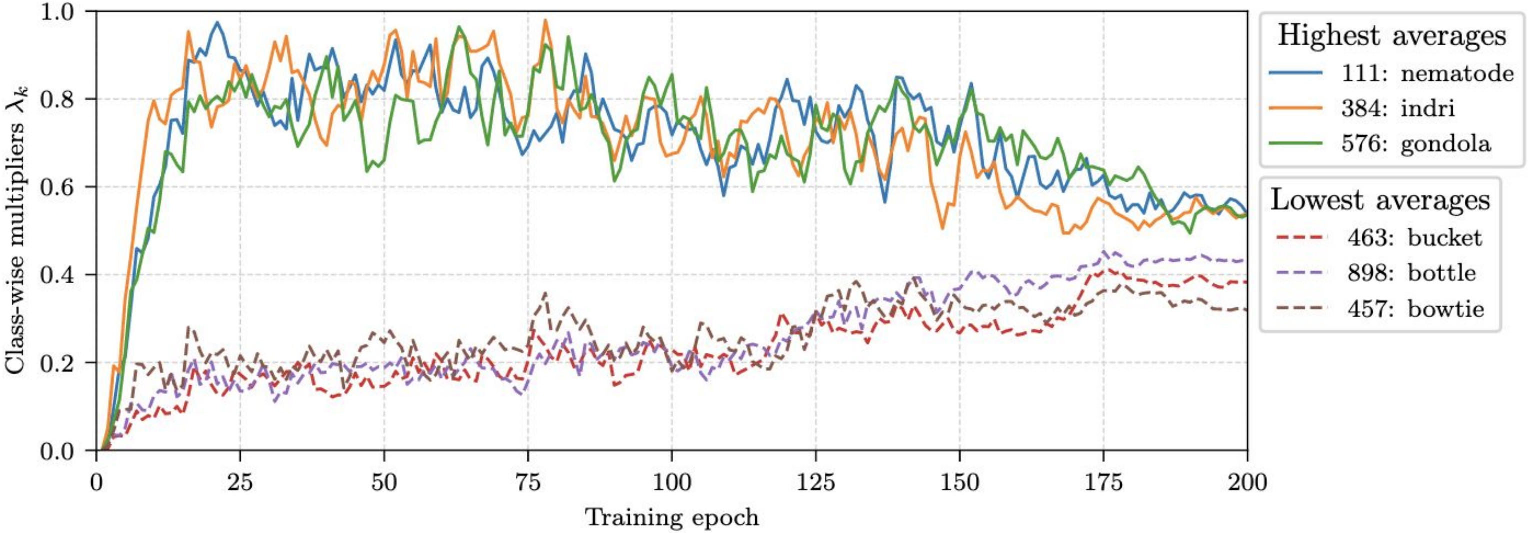


Figure 4. Visualization of learned multipliers λ_k during the training of the ResNet-50 model on ImageNet. We show classes with the highest average (*Solid lines*) and the lowest average (*dashed lines*).

Experiments

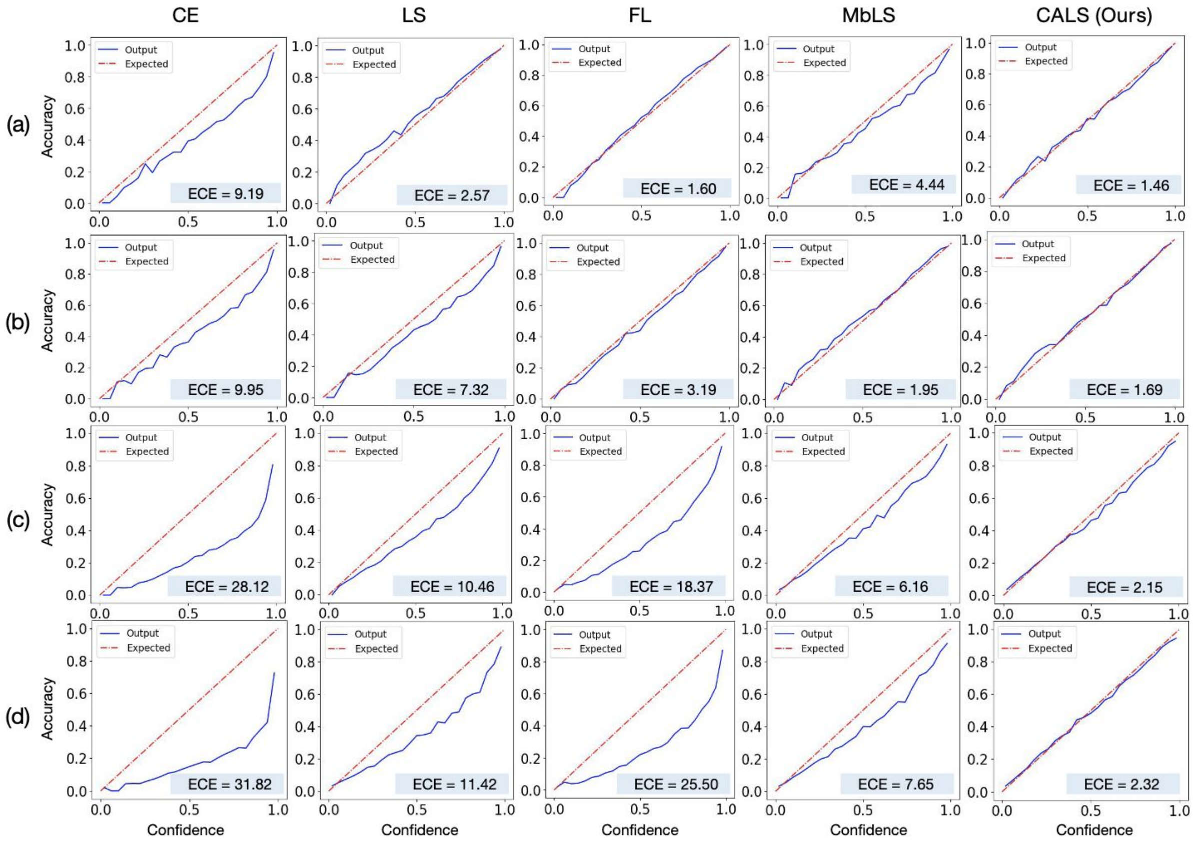


Figure 5. **Calibration visualizations: (a) ImageNet (ResNet-50), (b) ImageNet (SwinV2-T), (c) ImageNet-LT (ResNet-50), and (d) ImageNet-LT (SwinV2-T) .** We present the reliability diagrams of our method (CALS), compared with those of baselines and closely related works. The number of bins to plot reliability diagrams is set to 25.

Conclusions

- Propose a modified Augmented Lagrangian Multiplier method for calibrating DNNs, where adaptive class-wise multipliers are introduced.

- Demonstrate that the optimizer matters significantly for model calibration, and encourage more future research in this direction

Thank you !

Code : https://github.com/by-liu/CALS

Bingyuan Liu : https://by-liu.github.io