AdaFocal:
Calibration-aware Adaptive Focal Loss

Arindam Ghosh[1], Thomas Schaaf[1], Matt Gormley[2]

[1] 3M Health Information Systems

[2] Carnegie Mellon University

Motivation for Calibration

Motivation for Calibration

1. Neural networks have grown complex and larger achieving SOTA performance in every field.

Motivation for Calibration

1. Neural networks have grown complex and larger achieving SOTA performance in every field.

2. However, these networks suffer from poor calibration i.e., the confidence scores of the predictions do not reflect the real-world probabilities of those predictions being true.

Motivation for Calibration

1. Neural networks have grown complex and larger achieving SOTA performance in every field.

2. However, these networks suffer from poor calibration i.e., the confidence scores of the predictions do not reflect the real-world probabilities of those predictions being true.

3. This is of great concern, particularly for mission-critical applications (autonomous driving or medical diagnosis), wherein the downstream decision relies not only on the predictions but also on their confidences.

Motivation for Calibration

1. Neural networks have grown complex and larger achieving SOTA performance in every field.

2. However, these networks suffer from poor calibration i.e., the confidence scores of the predictions do not reflect the real-world probabilities of those predictions being true.

3. This is of great concern, particularly for mission-critical applications (autonomous driving or medical diagnosis), wherein the downstream decision relies not only on the predictions but also on their confidences.

Calibration: A network is said to be perfectly calibrated if the confidence score matches the probability of the model classifying the input correctly i.e.,

$$\mathbb{P}(\hat{y} = y_{\text{true}} \mid \hat{p}_{\hat{y}} = p) = p,$$

Motivation for Calibration

1. Neural networks have grown complex and larger achieving SOTA performance in every field.

2. However, these networks suffer from poor calibration i.e., the confidence scores of the predictions do not reflect the real-world probabilities of those predictions being true.

3. This is of great concern, particularly for mission-critical applications (autonomous driving or medical diagnosis), wherein the downstream decision relies not only on the predictions but also on their confidences.

Calibration: A network is said to be perfectly calibrated if the confidence score matches the probability of the model classifying the input correctly i.e.,

$$\mathbb{P}(\hat{y} = y_{\text{true}} \mid \hat{p}_{\hat{y}} = p) = p$$

If the network assigns a confidence of 0.8 to a set of predictions, we should expect 80% of those predictions to be correct.

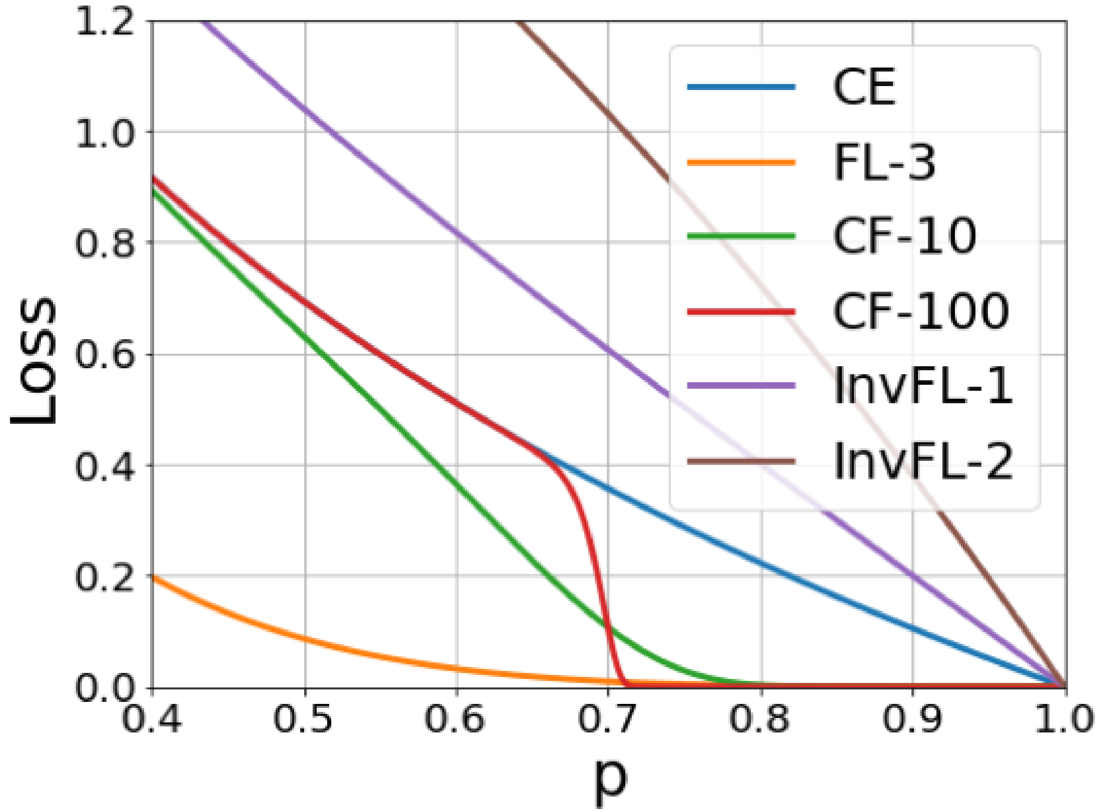
Calibration Properties of Focal Loss

Focal loss: $\mathcal{L}_{FL}(p) = -(1-p)^{\gamma} \log p$

Calibration Properties of Focal Loss

Focal loss: $\mathcal{L}_{FL}(p) = -(1-p)^{\gamma} \log p$

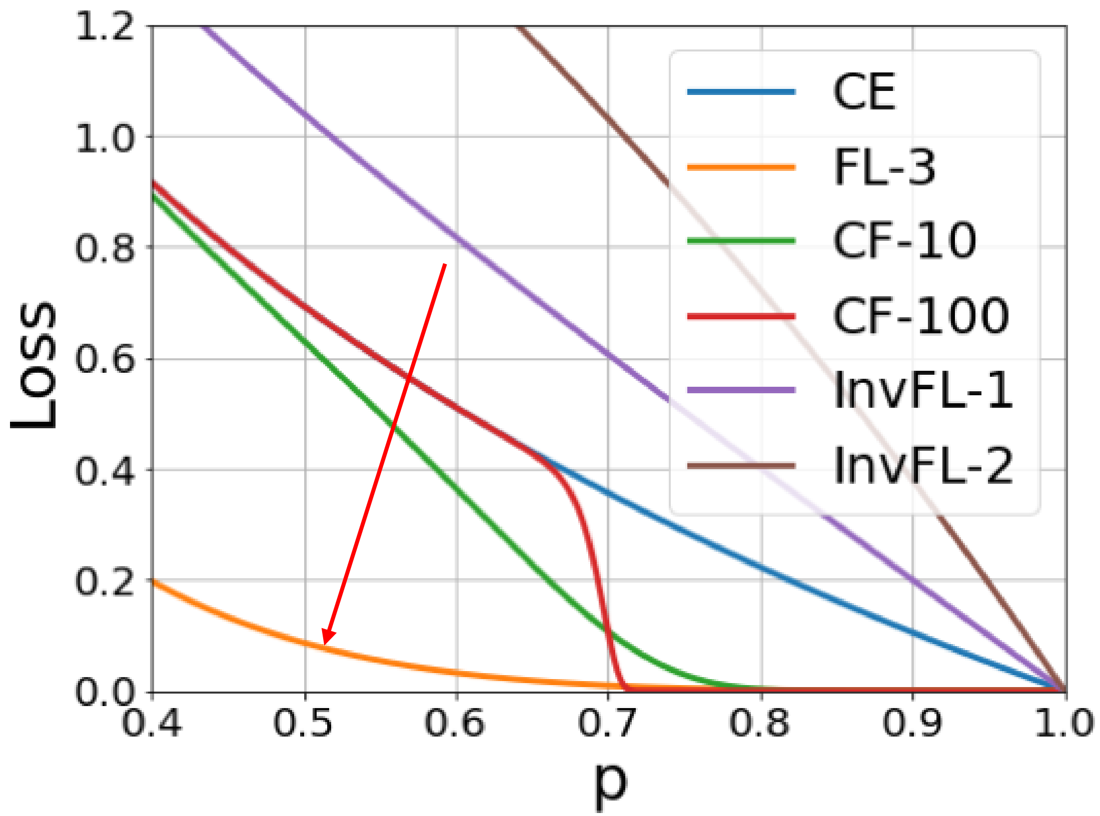
was originally proposed to improve the accuracy of classifiers by focusing on hard examples.



Calibration Properties of Focal Loss

Focal loss: $\mathcal{L}_{FL}(p) = -(1-p)^{\gamma} \log p$

was originally proposed to improve the accuracy of classifiers by focusing on hard examples.

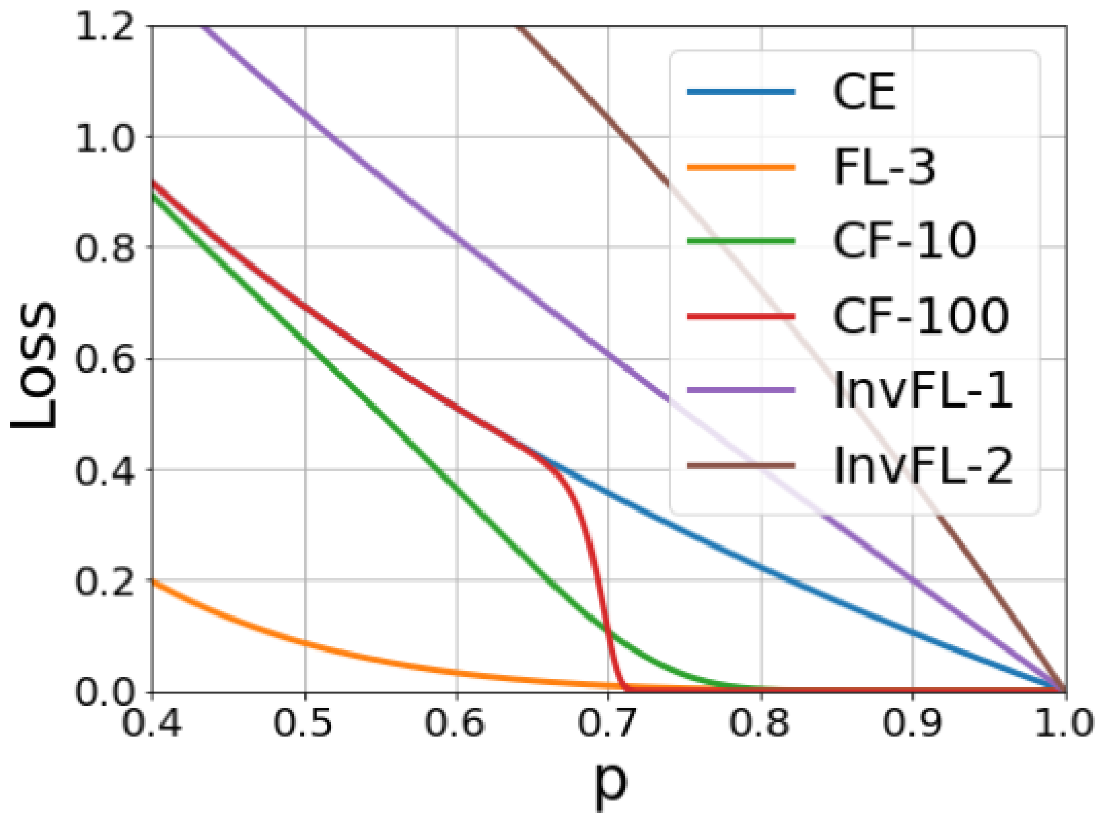


Calibration Properties of Focal Loss

Focal loss: $\mathcal{L}_{FL}(p) = -(1-p)^{\gamma} \log p$

was originally proposed to improve the accuracy of classifiers by focusing on hard examples.

Recently, it was shown that training with focal loss results in better calibration than cross entropy [1].



[1] Mukhoti et al., Calibrating deep neural networks using focal loss. In NeurIPS 2020.

Calibration Properties of Focal Loss

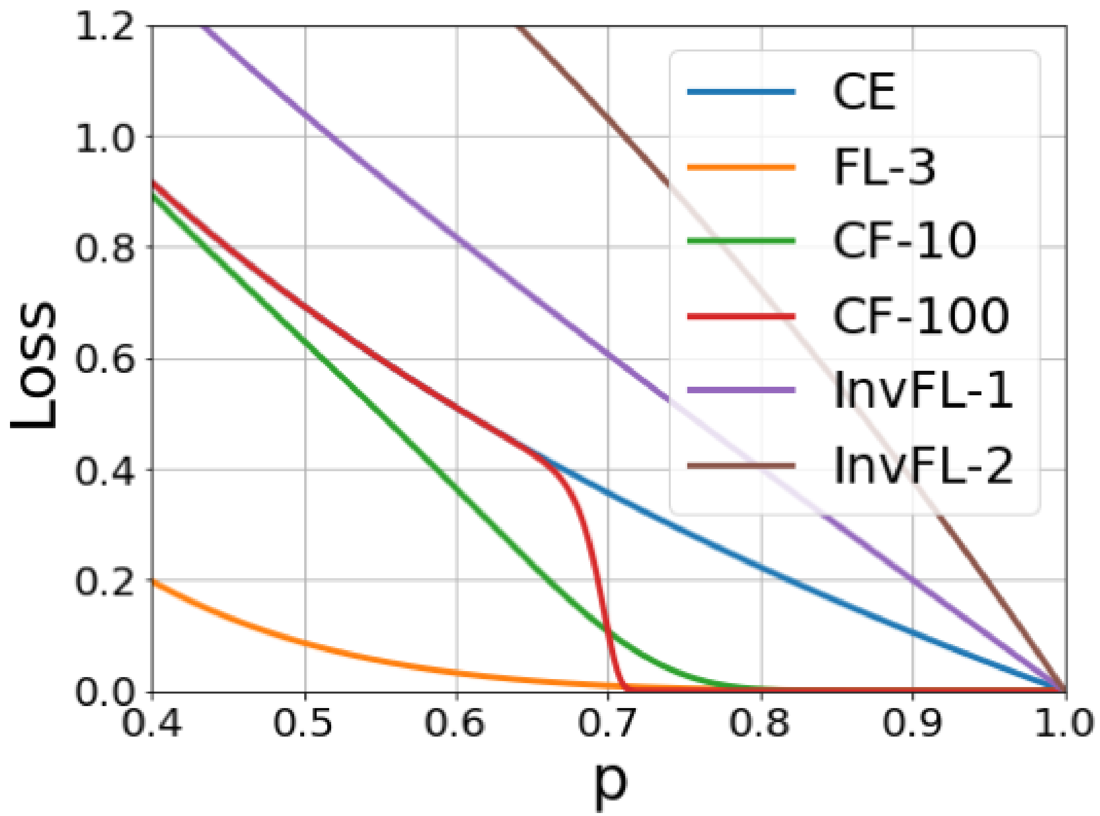
Focal loss: $\mathcal{L}_{FL}(p) = -(1-p)^\gamma \log p$

was originally proposed to improve the accuracy of classifiers by focusing on hard examples.

Recently, it was shown that training with focal loss results in better calibration than cross entropy [1].

This is because, focal loss, while minimizing the KL divergence, increases the entropy of the prediction (using the parameter γ) to counter over-confidence.

$$\mathcal{L}_{FL} \geq KL(q||\hat{\mathbf{p}}) - \gamma\mathbb{H}(\hat{\mathbf{p}})$$



[1] Mukhoti et al., Calibrating deep neural networks using focal loss. In NeurIPS 2020.

Calibration Properties of Focal Loss

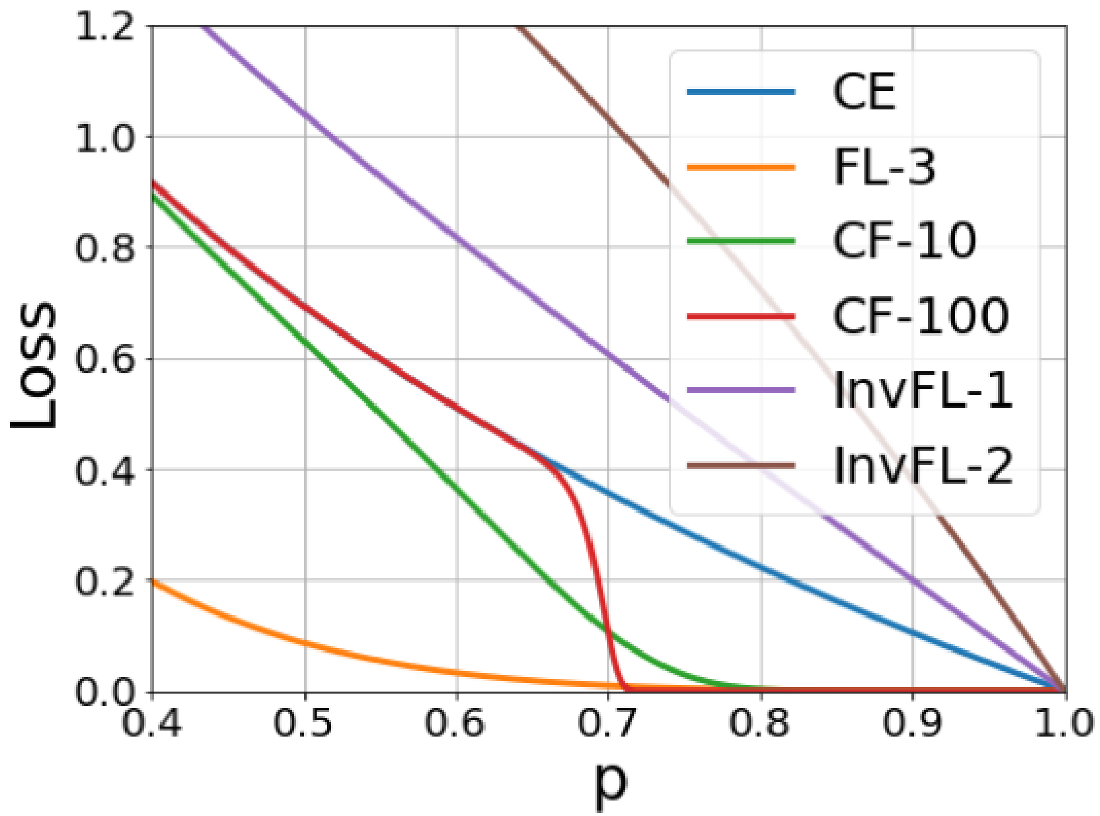
Focal loss: $\mathcal{L}_{FL}(p) = -(1-p)^{\gamma} \log p$

was originally proposed to improve the accuracy of classifiers by focusing on hard examples.

Recently, it was shown that training with focal loss results in better calibration than cross entropy [1].

This is because, focal loss, while minimizing the KL divergence, increases the entropy of the prediction (using the parameter γ) to counter over-confidence.

$$\mathcal{L}_{FL} \geq KL(q||\hat{\mathbf{p}}) - \gamma\mathbb{H}(\hat{\mathbf{p}})$$

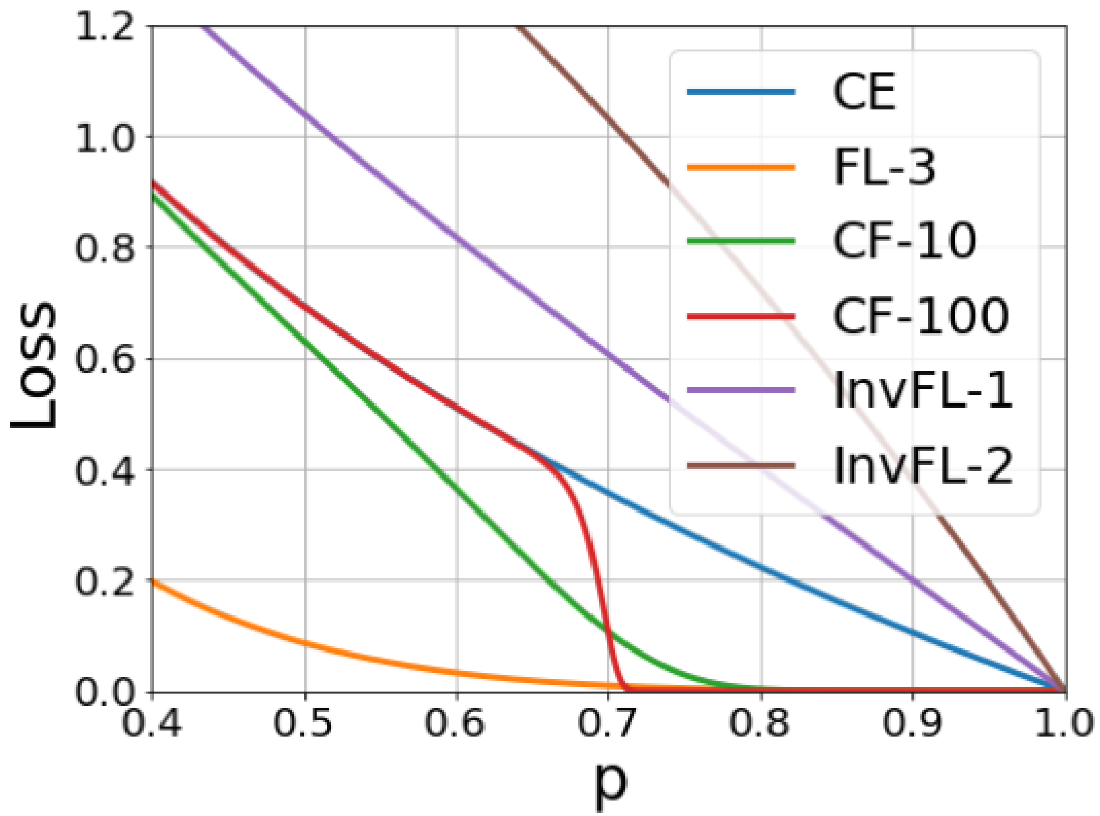


[1] Mukhoti et al., Calibrating deep neural networks using focal loss. In NeurIPS 2020.

Calibration Properties of Focal Loss

Focal loss: $\mathcal{L}_{FL}(p) = -(1-p)^{\gamma}\log p$

was originally proposed to improve the accuracy of classifiers by focusing on hard examples.



Recently, it was shown that training with focal loss results in better calibration than cross entropy [1].

This is because, focal loss, while minimizing the KL divergence, increases the entropy of the prediction (using the parameter γ) to counter over-confidence.

$$\mathcal{L}_{FL} \geq KL(q||\hat{\mathbf{p}}) - \gamma\mathbb{H}(\hat{\mathbf{p}})$$

Inverse Focal loss: $\mathcal{L}_{InvFL}(p) = -(1+p)^{\gamma}\log p,$

[1] Mukhoti et al., Calibrating deep neural networks using focal loss. In NeurIPS 2020.

Calibration Properties of Focal Loss

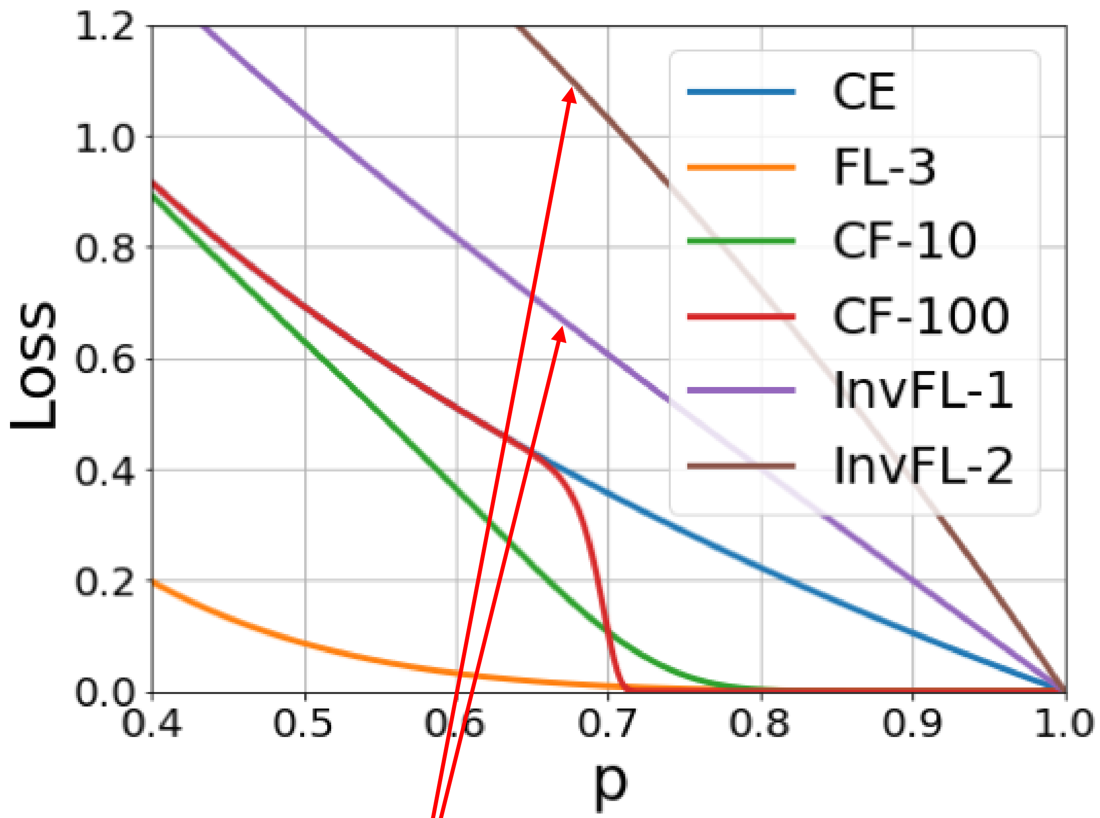
Focal loss: $\mathcal{L}_{FL}(p) = -(1-p)^{\gamma} \log p$

was originally proposed to improve the accuracy of classifiers by focusing on hard examples.

Recently, it was shown that training with focal loss results in better calibration than cross entropy [1].

This is because, focal loss, while minimizing the KL divergence, increases the entropy of the prediction (using the parameter γ) to counter over-confidence.

$$\mathcal{L}_{FL} \geq KL(q||\hat{\mathbf{p}}) - \gamma \mathbb{H}(\hat{\mathbf{p}})$$



Inverse Focal loss: $\mathcal{L}_{InvFL}(p) = -(1+p)^{\gamma} \log p,$

[1] Mukhoti et al., Calibrating deep neural networks using focal loss. In NeurIPS 2020.

Calibration Properties of Focal Loss

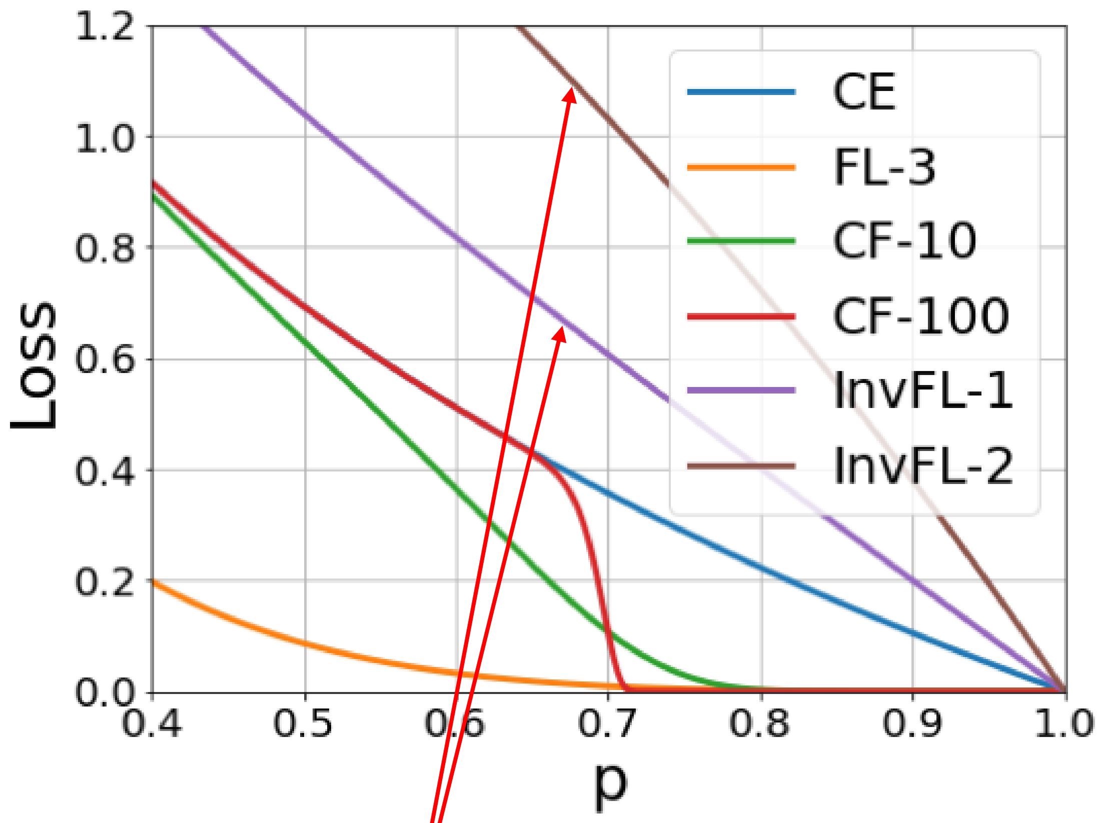
Focal loss: $\mathcal{L}_{FL}(p) = -(1-p)^{\gamma} \log p$

was originally proposed to improve the accuracy of classifiers by focusing on hard examples.

Recently, it was shown that training with focal loss results in better calibration than cross entropy [1].

This is because, focal loss, while minimizing the KL divergence, increases the entropy of the prediction (using the parameter γ) to counter over-confidence.

$$\mathcal{L}_{FL} \geq KL(q||\hat{\mathbf{p}}) - \gamma \mathbb{H}(\hat{\mathbf{p}})$$



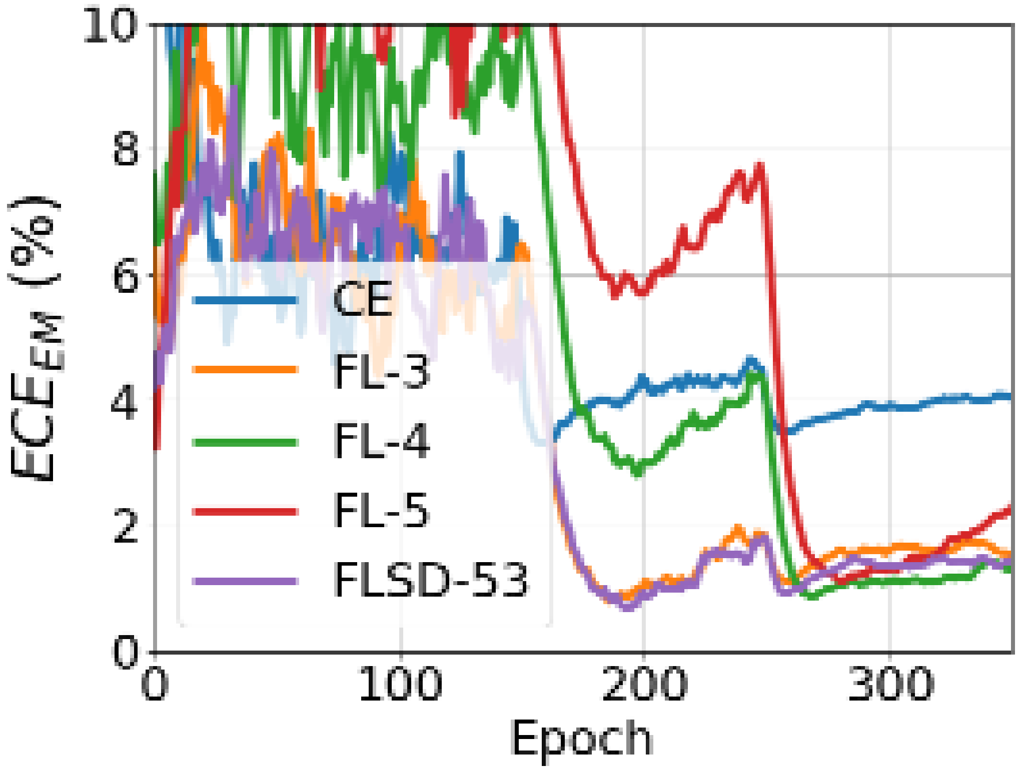
Inverse Focal loss: $\mathcal{L}_{InvFL}(p) = -(1+p)^{\gamma} \log p,$

serves the opposite purpose of focal loss. It helps recover from under-confidence by pushing the confidence scores even higher.

[1] Mukhoti et al., Calibrating deep neural networks using focal loss. In NeurIPS 2020.

Limitations of Focal Loss (with fixed γ)

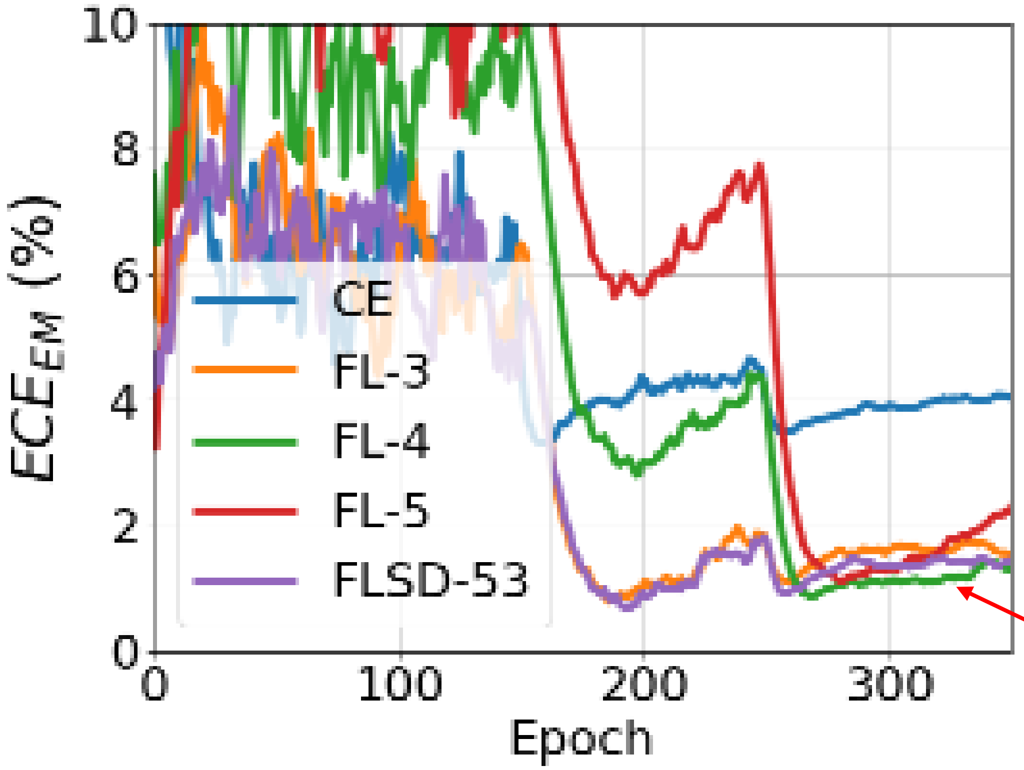
ResNet-50 trained on CIFAR-10



(a) $\mathrm{ECE}_{\mathrm{EM}}$ (%)

Limitations of Focal Loss (with fixed γ)

ResNet-50 trained on CIFAR-10

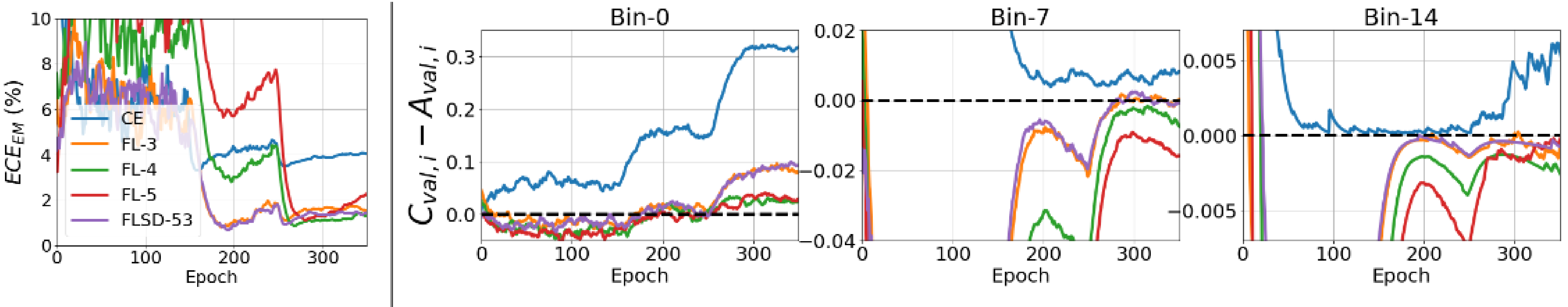


(a) $\mathrm{ECE}_{\mathrm{EM}}$ (%)

Focal loss γ = 4 achieves the overall lowest ECE_EM.

Limitations of Focal Loss (with fixed γ)

ResNet-50 trained on CIFAR-10



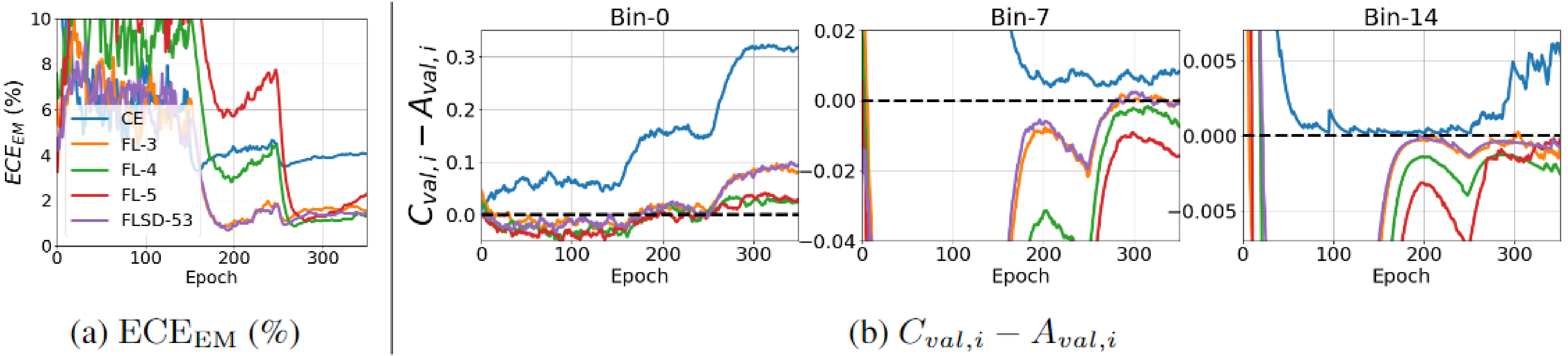
(a) $\mathrm{ECE}_{\mathrm{EM}}$ (%)

(b) $C_{val,i} - A_{val,i}$

Focal loss γ = 4 achieves the overall lowest ECE_EM.

Limitations of Focal Loss (with fixed γ)

ResNet-50 trained on CIFAR-10



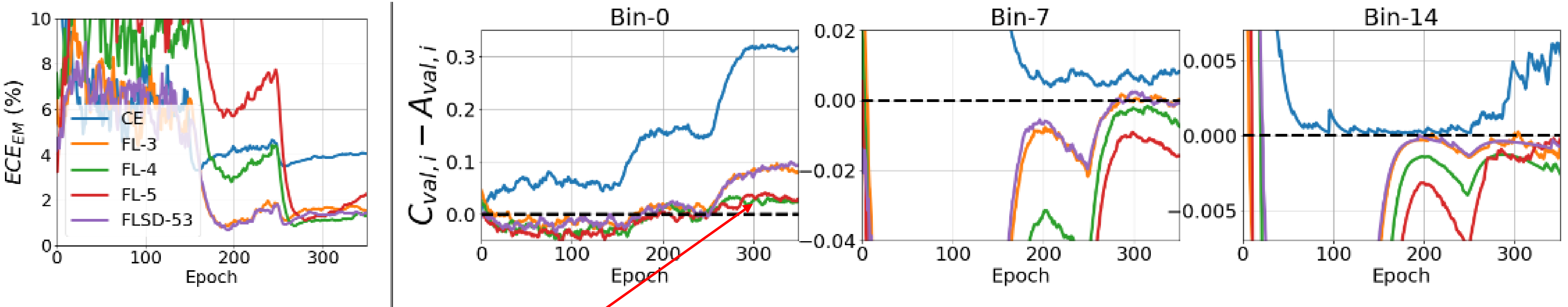
(a) $\mathrm{ECE}_{\mathrm{EM}}$ (%)

(b) $C_{val,i} - A_{val,i}$

Focal loss γ = 4 achieves the overall lowest ECE_EM.
However, there's no single γ that performs the best across all the bins.

Limitations of Focal Loss (with fixed γ)

ResNet-50 trained on CIFAR-10



(a) $\mathrm{ECE_{EM}}$ (%)

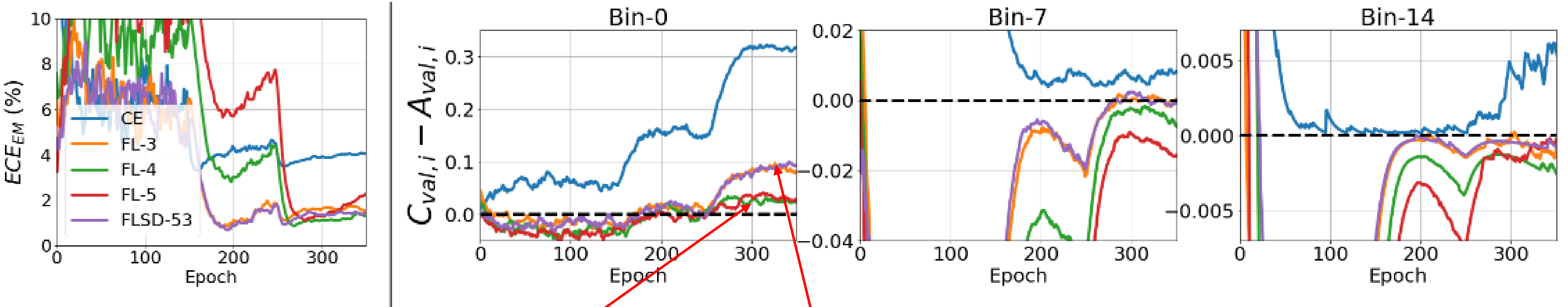
(b) $C_{val,i} - A_{val,i}$

Focal loss γ = 4 achieves the overall lowest ECE_EM.
However, there's no single γ that performs the best across all the bins.

For example,
In bin-0, γ = 4, 5 achieves better calibration whereas γ = 0, 3 are over-confident.

Limitations of Focal Loss (with fixed γ)

ResNet-50 trained on CIFAR-10



(a) $\mathrm{ECE}_{\mathrm{EM}}$ (%)

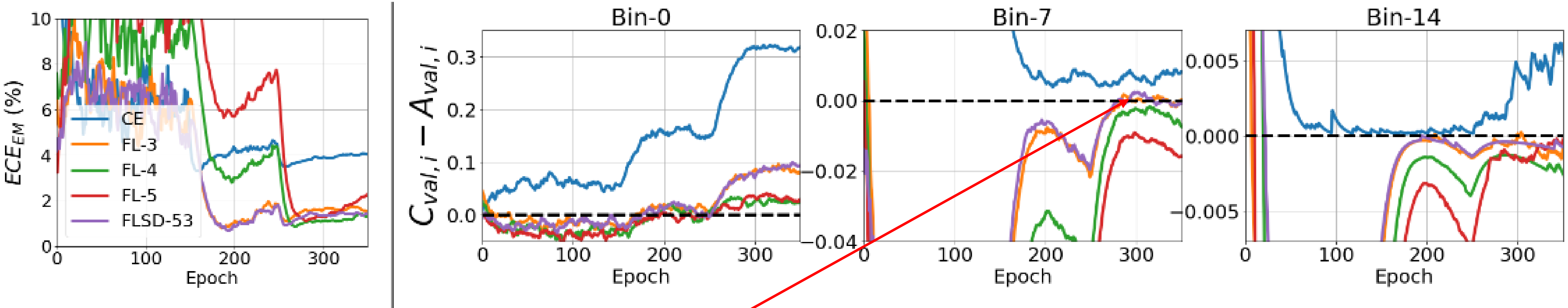
(b) $C_{val,i} - A_{val,i}$

Focal loss γ = 4 achieves the overall lowest ECE_EM.
However, there's no single γ that performs the best across all the bins.

For example,
In bin-0, γ = 4, 5 achieves better calibration whereas γ = 0, 3 are over-confident.

Limitations of Focal Loss (with fixed γ)

ResNet-50 trained on CIFAR-10



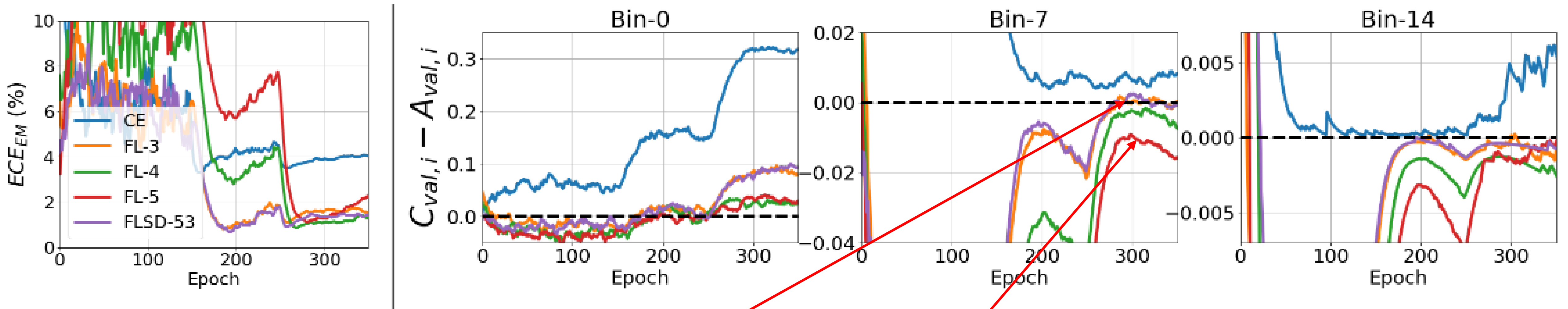
(a) $\mathrm{ECE}_{\mathrm{EM}}$ (%)　(b) $C_{val,i} - A_{val,i}$

Focal loss γ = 4 achieves the overall lowest ECE_EM.
However, there's no single γ that performs the best across all the bins.

For example,
In bin-0, γ = 4, 5 achieves better calibration whereas γ = 0, 3 are over-confident.
In bin-7, γ = 3 is better calibrated, whereas γ = 4, 5 are under-confident and γ = 0 is over-confident.

Limitations of Focal Loss (with fixed γ)

ResNet-50 trained on CIFAR-10



(a) $\mathrm{ECE}_{\mathrm{EM}}$ (%)

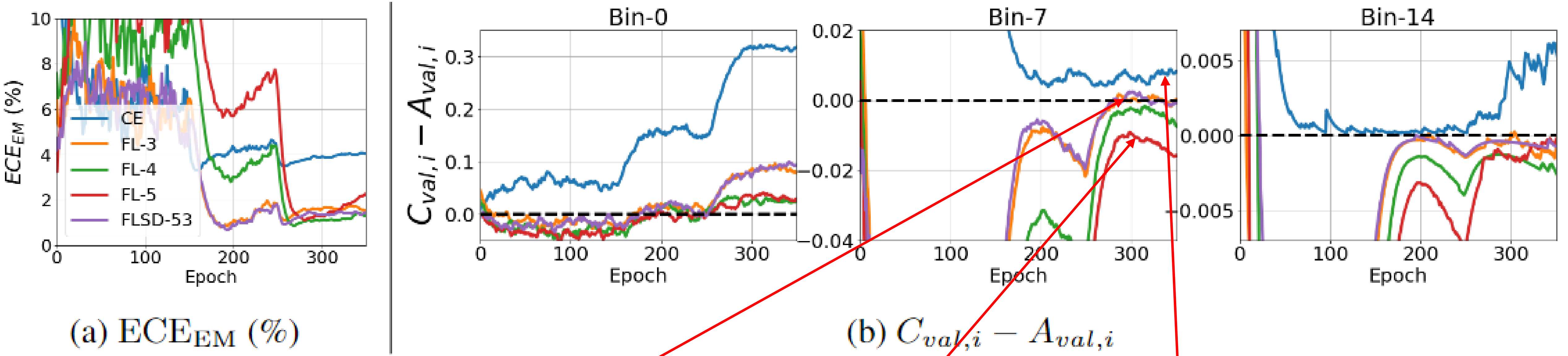
(b) $C_{val,i} - A_{val,i}$

Focal loss γ = 4 achieves the overall lowest ECE_EM.
However, there's no single γ that performs the best across all the bins.

For example,
In bin-0, γ = 4, 5 achieves better calibration whereas γ = 0, 3 are over-confident.
In bin-7, γ = 3 is better calibrated, whereas γ = 4, 5 are under-confident and γ = 0 is over-confident.

Limitations of Focal Loss (with fixed γ)

ResNet-50 trained on CIFAR-10



(a) $\mathrm{ECE}_{\mathrm{EM}}$ (%)   (b) $C_{val,i} - A_{val,i}$

Focal loss γ = 4 achieves the overall lowest ECE_EM.
However, there's no single γ that performs the best across all the bins.

For example,
In bin-0, γ = 4, 5 achieves better calibration whereas γ = 0, 3 are over-confident.
In bin-7, γ = 3 is better calibrated, whereas γ = 4, 5 are under-confident and γ = 0 is over-confident.

Motivation for Focal loss with adaptive γ (AdaFocal)

This motivates the design of a training strategy that can assign an appropriate γ for each bin.

Motivation for Focal loss with adaptive γ (AdaFocal)

This motivates the design of a training strategy that can assign an appropriate γ for each bin.

Challenges:

Motivation for Focal loss with adaptive γ (AdaFocal)

This motivates the design of a training strategy that can assign an appropriate γ for each bin.

Challenges:

1. How do we find a correspondence between the confidence of training samples (which we can manipulate during training using the parameter γ) and the confidence of the validation or test samples (which are our actual targets, but we do not have direct control over)?

Motivation for Focal loss with adaptive γ (AdaFocal)

This motivates the design of a training strategy that can assign an appropriate γ for each bin.

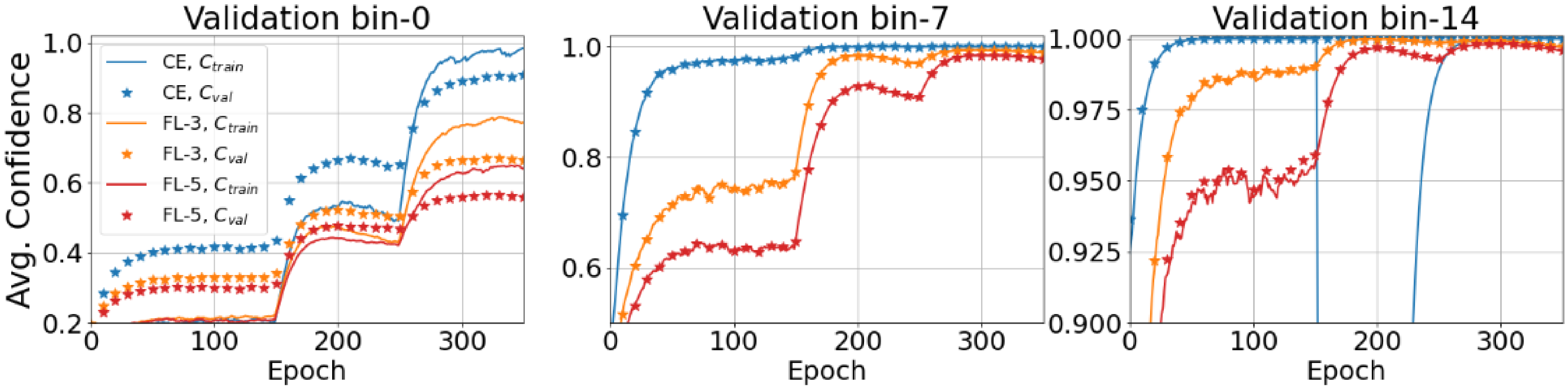
Challenges:

1. How do we find a correspondence between the confidence of training samples (which we can manipulate during training using the parameter γ) and the confidence of the validation or test samples (which are our actual targets, but we do not have direct control over)?

2. Given some correspondence, how do we arrive at the appropriate values of γ that will lead to the best calibration?

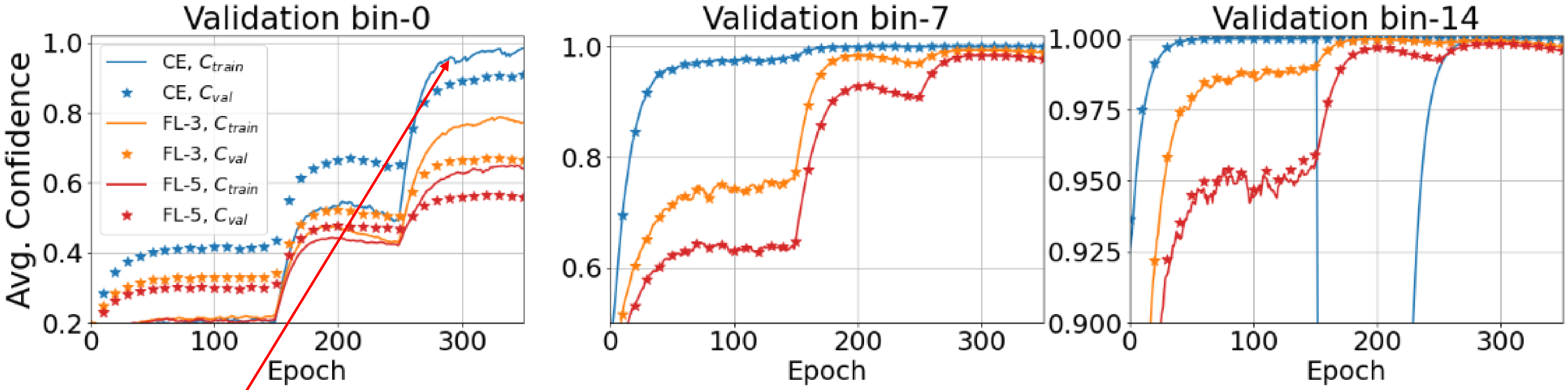
Correspondence between Confidence of Training and Validation Samples

Correspondence between Confidence of Training and Validation Samples



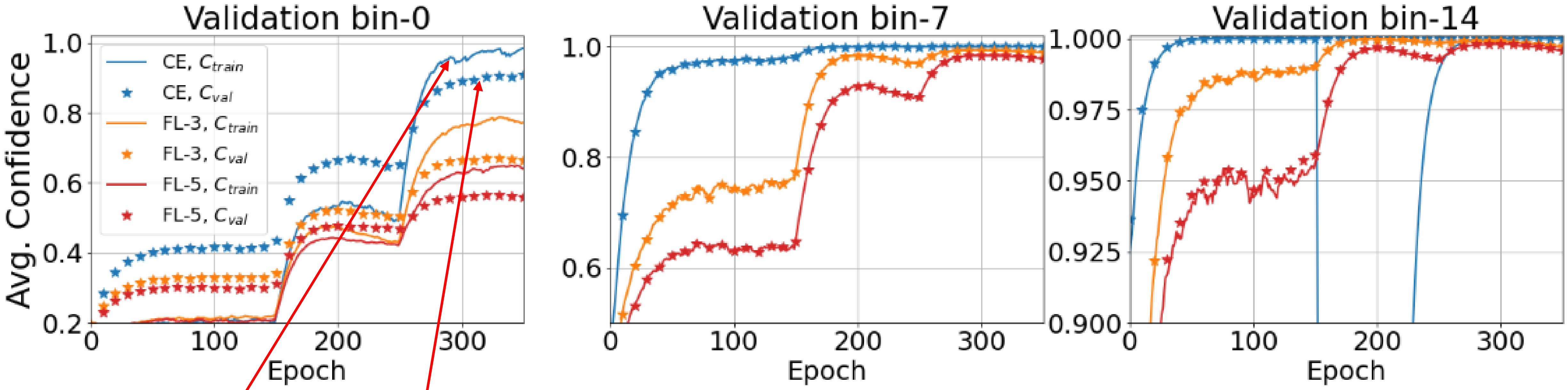
Compare C_train with C_val (in a lower, middle and higher probability region/bin) and find that there is indeed a good correspondence between the two quantities.

Correspondence between Confidence of Training and Validation Samples



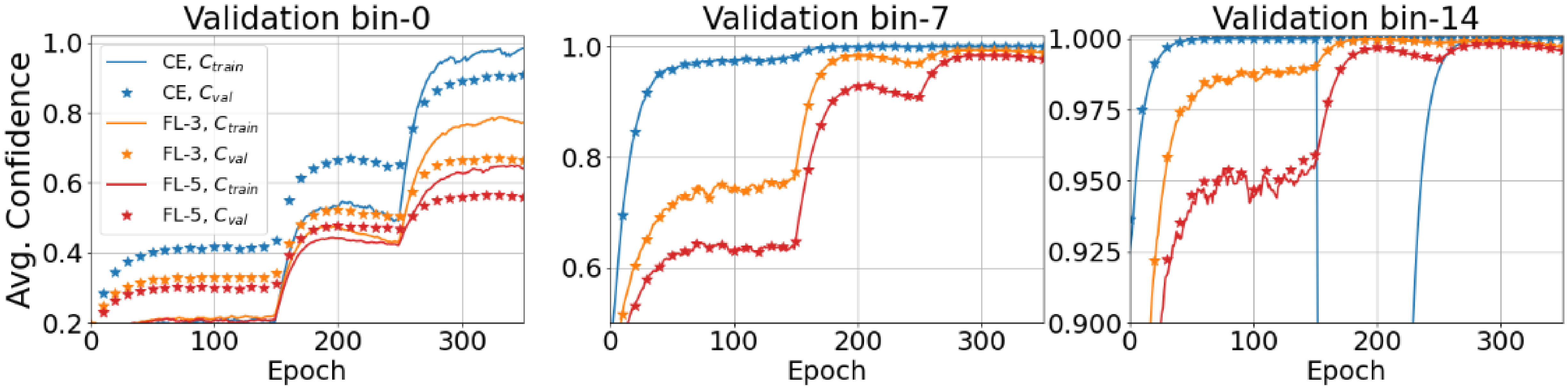
Compare C_train with C_val (in a lower, middle and higher probability region/bin) and find that there is indeed a good correspondence between the two quantities.

Correspondence between Confidence of Training and Validation Samples



Compare C_train with C_val (in a lower, middle and higher probability region/bin) and find that there is indeed a good correspondence between the two quantities.

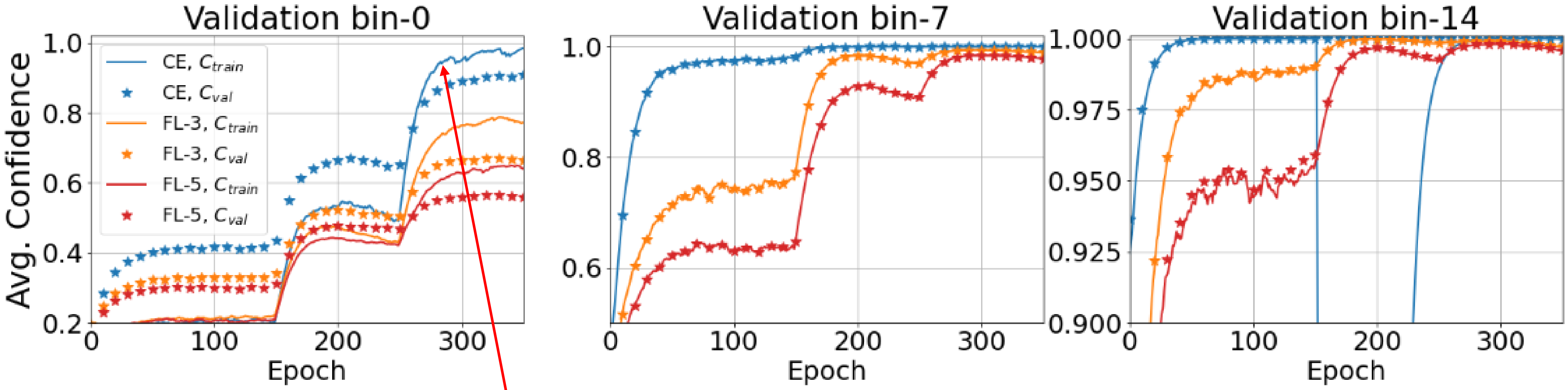
Correspondence between Confidence of Training and Validation Samples



Compare C_train with C_val (in a lower, middle and higher probability region/bin) and find that there is indeed a good correspondence between the two quantities.

For example, as γ increases from 0 to 3, to 5, the solid-line C_train gets lower, and the same behavior is observed for the starred-line C_val.

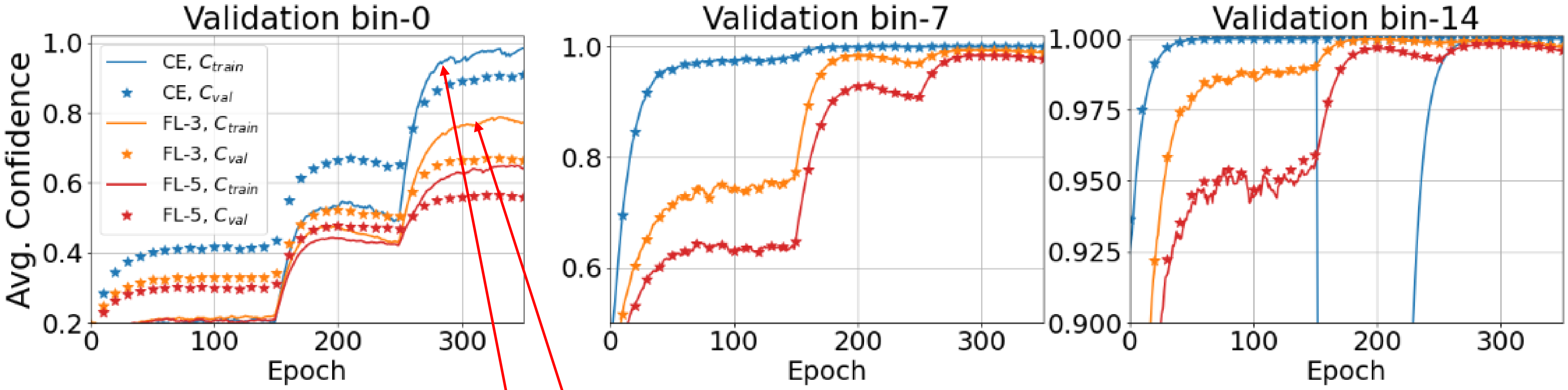
Correspondence between Confidence of Training and Validation Samples



Compare C_train with C_val (in a lower, middle and higher probability region/bin) and find that there is indeed a good correspondence between the two quantities.

For example, as γ increases from 0 to 3, to 5, the solid-line C_train gets lower, and the same behavior is observed for the starred-line C_val.

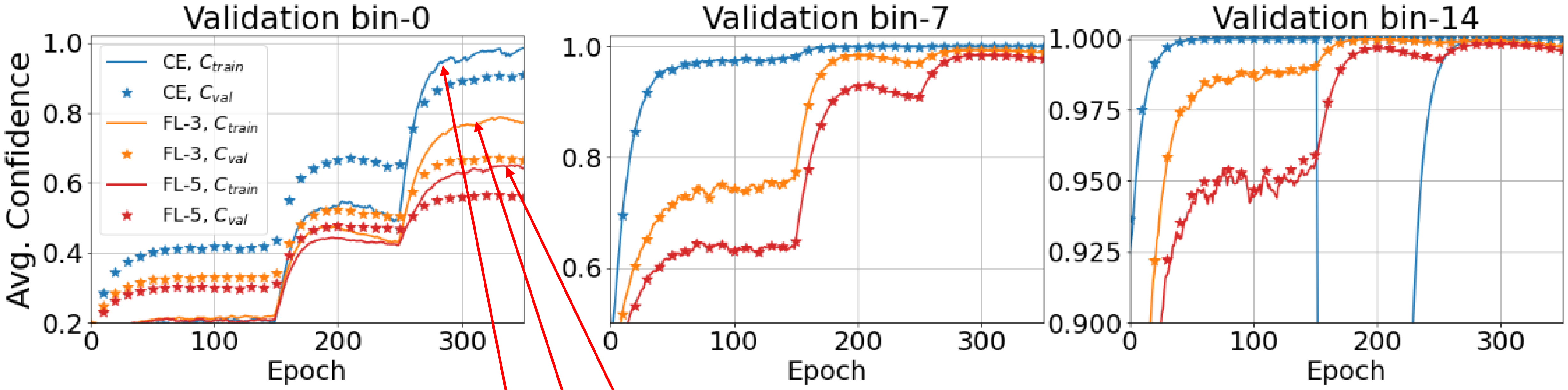
Correspondence between Confidence of Training and Validation Samples



Compare C_train with C_val (in a lower, middle and higher probability region/bin) and find that there is indeed a good correspondence between the two quantities.

For example, as γ increases from 0 to 3, to 5, the solid-line C_train gets lower, and the same behavior is observed for the starred-line C_val.

Correspondence between Confidence of Training and Validation Samples



Compare C_train with C_val (in a lower, middle and higher probability region/bin) and find that there is indeed a good correspondence between the two quantities.

For example, as γ increases from 0 to 3, to 5, the solid-line C_train gets lower, and the same behavior is observed for the starred-line C_val.

Proposed AdaFocal loss and Gamma-update rule:

Proposed AdaFocal loss and Gamma-update rule:

Based on the correspondence, the proposed **AdaFocal loss function** is given by

$$\mathcal{L}_{AdaFocal}(p_n, t) = \begin{cases} -(1-p_n)^{\gamma_{t,b}} \log p_n, & \text{if } \gamma_{t,b} \geq 0 \\ -(1+p_n)^{|\gamma_{t,b}|} \log p_n, & \text{if } \gamma_{t,b} < 0, \end{cases}$$

Proposed AdaFocal loss and Gamma-update rule:

Based on the correspondence, the proposed **AdaFocal loss function** is given by

$$\mathcal{L}_{AdaFocal}(p_n, t) = \begin{cases} -(1-p_n)^{\gamma_{t,b}} \log p_n, & \text{if } \gamma_{t,b} \geq 0 \\ -(1+p_n)^{|\gamma_{t,b}|} \log p_n, & \text{if } \gamma_{t,b} < 0, \end{cases}$$

The loss function switches between Focal and inverse-Focal loss based on the value of γ in each bin.

Proposed AdaFocal loss and Gamma-update rule:

Based on the correspondence, the proposed **AdaFocal loss function** is given by

$$\mathcal{L}_{AdaFocal}(p_n, t) = \begin{cases} -(1-p_n)^{\gamma_{t,b}} \log p_n, & \text{if } \gamma_{t,b} \geq 0 \\ -(1+p_n)^{|\gamma_{t,b}|} \log p_n, & \text{if } \gamma_{t,b} < 0, \end{cases}$$

The loss function switches between Focal and inverse-Focal loss based on the value of γ in each bin.

The **gamma-update rule** is given by

$$\gamma_{t,b} = \gamma_{t-1,b} * \exp(\lambda(C_{val,b} - A_{val,b}))$$

Proposed AdaFocal loss and Gamma-update rule:

Based on the correspondence, the proposed **AdaFocal loss function** is given by

$$\mathcal{L}_{AdaFocal}(p_n, t) = \begin{cases} -(1-p_n)^{\gamma_{t,b}} \log p_n, & \text{if } \gamma_{t,b} \geq 0 \\ -(1+p_n)^{|\gamma_{t,b}|} \log p_n, & \text{if } \gamma_{t,b} < 0, \end{cases}$$

The loss function switches between Focal and inverse-Focal loss based on the value of γ in each bin.

The **gamma-update rule** is given by

$$\gamma_{t,b} = \gamma_{t-1,b} * \exp(\lambda(C_{val,b} - A_{val,b}))$$

which adaptively modifies γ based on γ_{t-1} from the previous time step and the magnitude of the mis-calibration C_val – A_val on the validation set.

Experiments:

Experiments:

Datasets (4 Image recognition and 1 text classification):

1. CIFAR-10
2. CIFAR-100
3. Tiny-ImageNet
4. ImageNet
5. 20 Newsgroup

Experiments:

Datasets (4 Image recognition and 1 text classification):

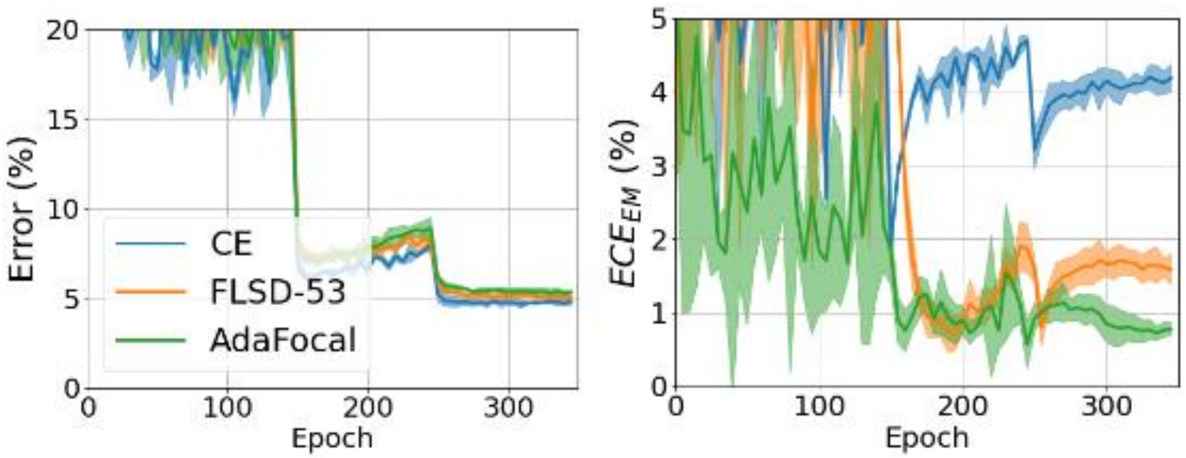
1. CIFAR-10
2. CIFAR-100
3. Tiny-ImageNet
4. ImageNet
5. 20 Newsgroup

Neural network architectures:

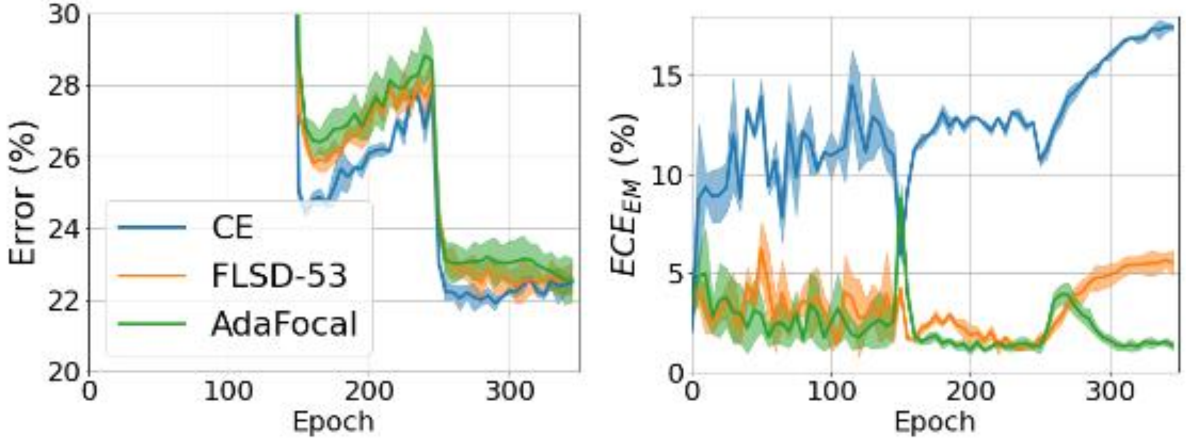
1. ResNet50, ResNet-100
2. Wide-ResNet-26-10
3. DenseNet-121
4. Global-pooling CNN
5. Pre-trained BERT

Results (Test set Error and ECE):

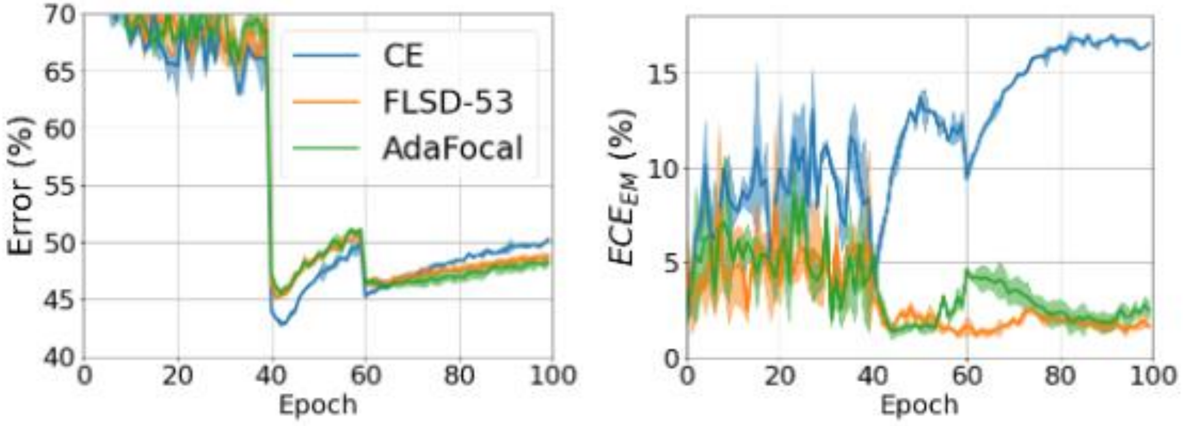
Results (Test set Error and ECE):



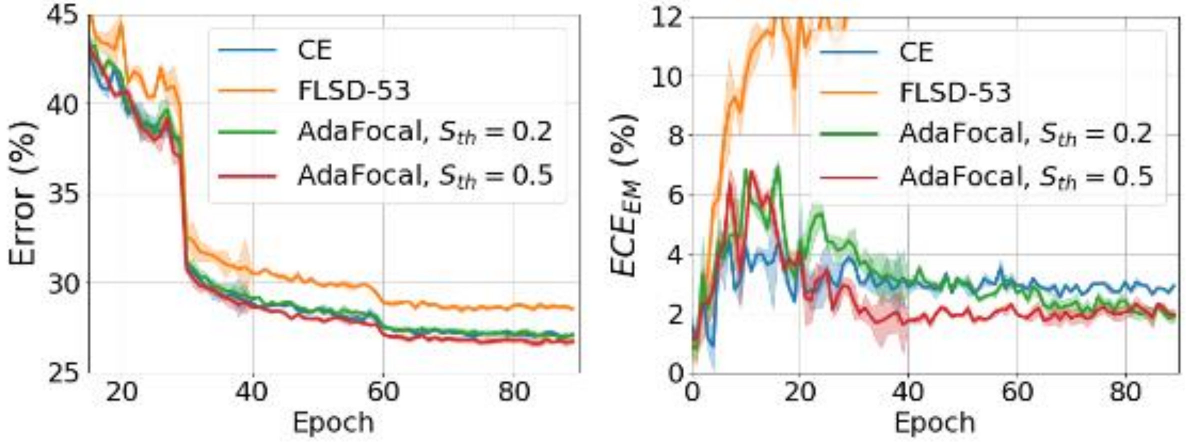
(a) CIFAR-10



(b) CIFAR-100

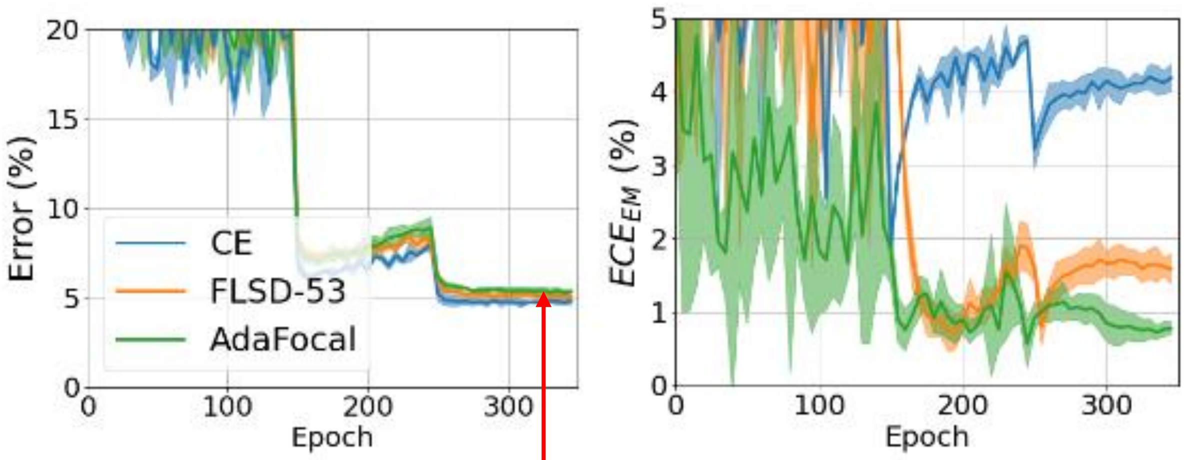


(c) Tiny-ImageNet

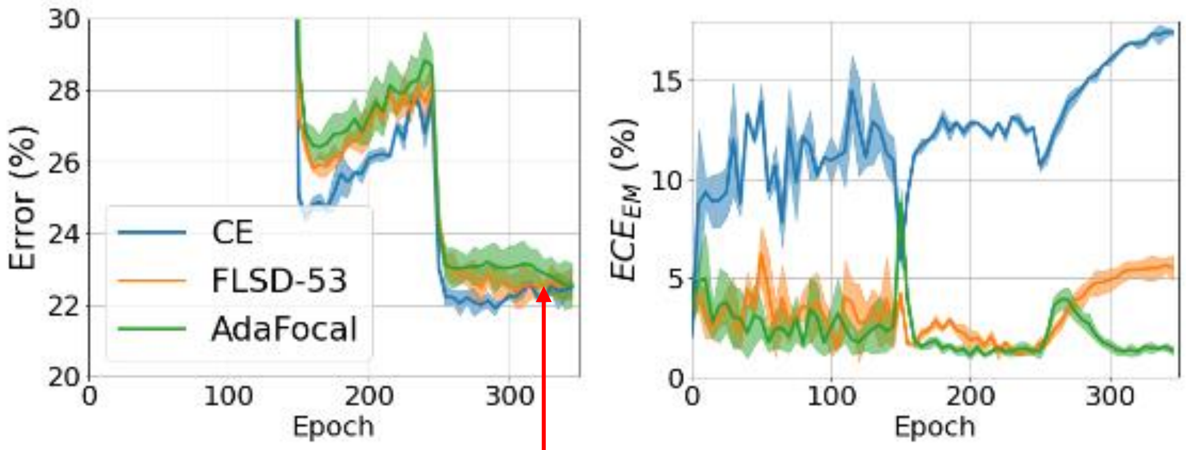


(d) ImageNet

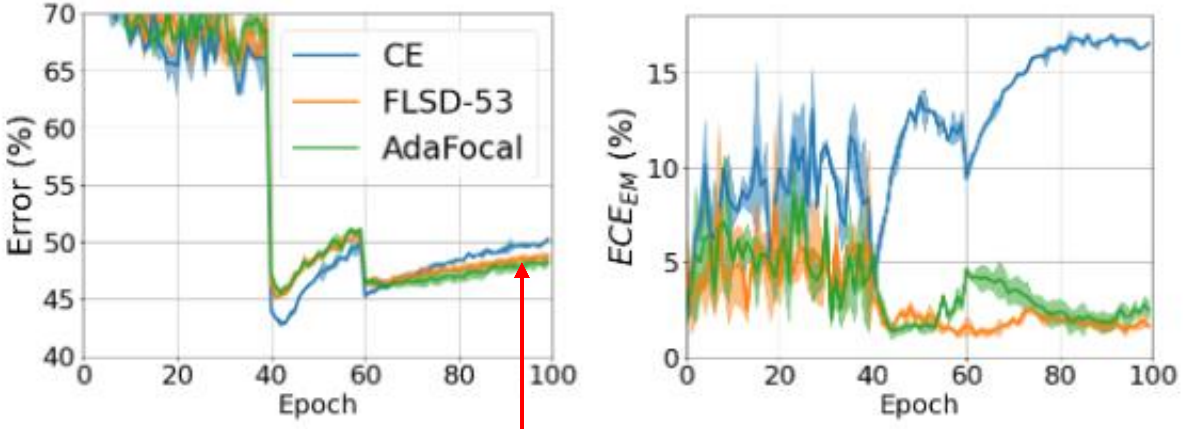
Results (Test set Error and ECE):



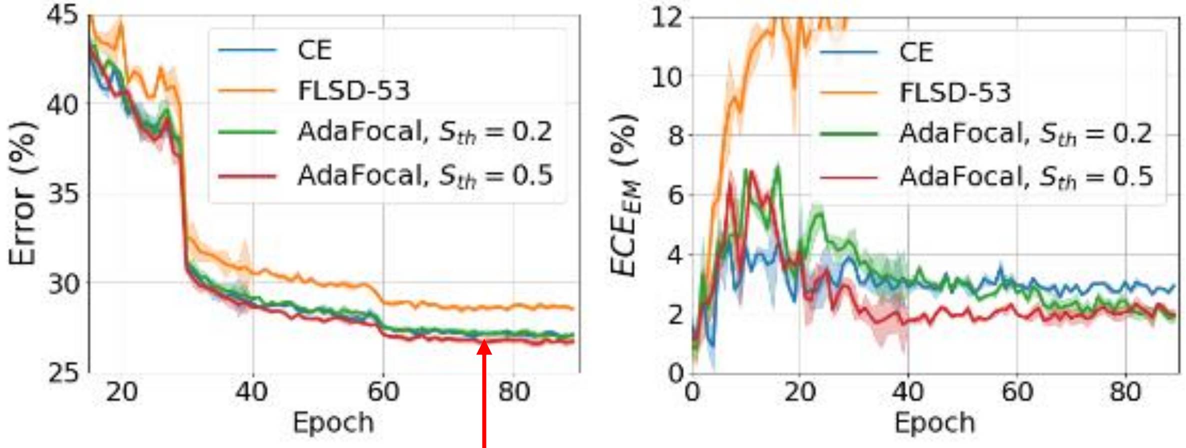
(a) CIFAR-10



(b) CIFAR-100

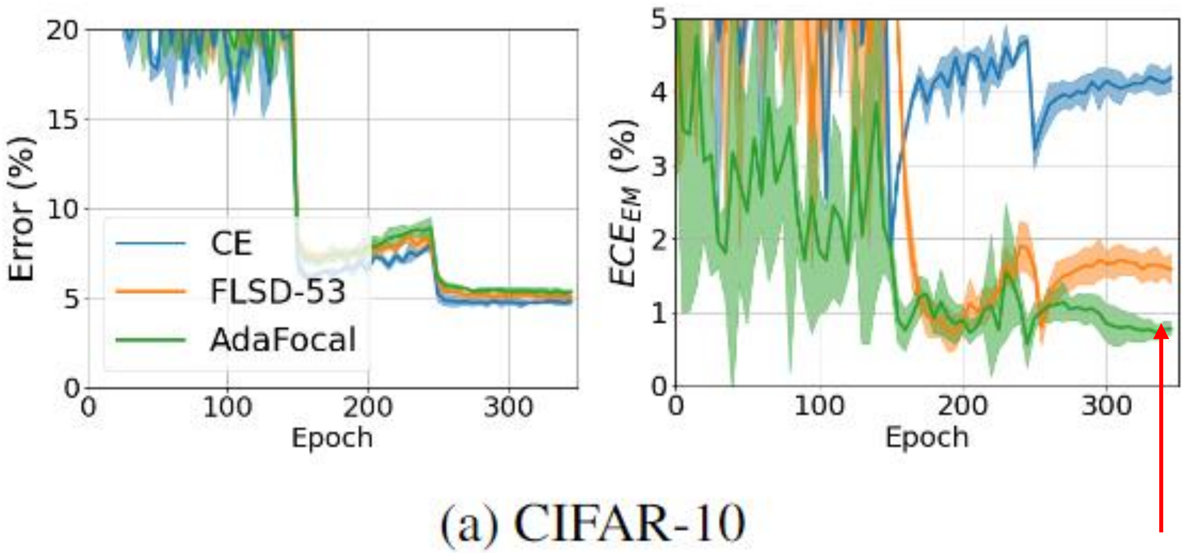


(c) Tiny-ImageNet

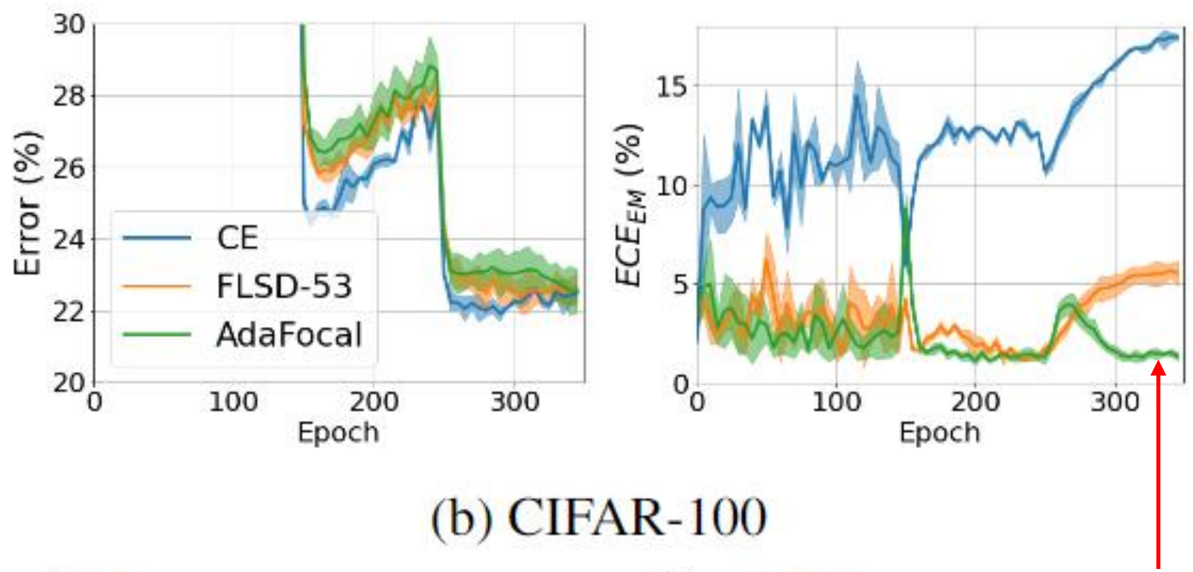


(d) ImageNet

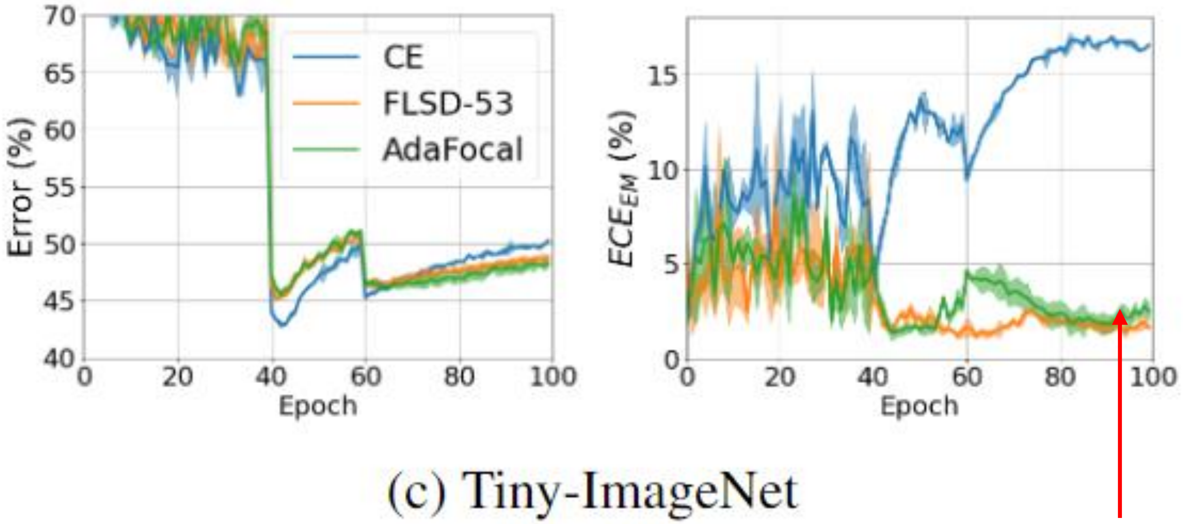
Results (Test set Error and ECE):



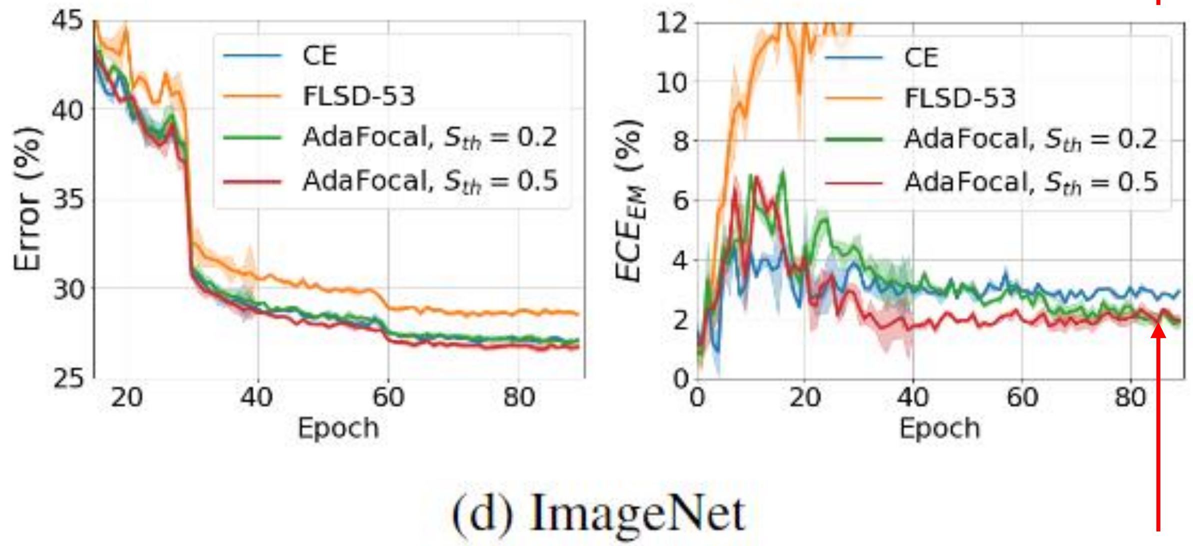
(a) CIFAR-10



(b) CIFAR-100



(c) Tiny-ImageNet



(d) ImageNet

Test Set ECE (all results)

Dataset	Model	Pre Temperature scaling						Post Temperature scaling					
		CE	Brier	MMCE	LS	FLSD-53	AdaFocal	CE	Brier	MMCE	LS	FLSD-53	AdaFocal
CIFAR-10	ResNet50	4.24	1.78	4.52	3.86	1.63	**0.66**	2.11(2.52)	1.24(1.11)	2.12(2.65)	2.97(0.92)	1.42(1.08)	**0.44(1.06)**
	ResNet110	4.39	2.63	5.16	4.44	1.90	**0.71**	2.27(2.74)	1.75(1.21)	2.53(2.83)	4.44(1.00)	1.25(1.20)	**0.73(1.02)**
	WideResNet	3.42	1.72	3.31	4.26	1.82	**0.64**	1.87(2.16)	1.72(1.00)	1.6(2.22)	2.44(0.81)	1.57(0.94)	**0.44(1.06)**
	DenseNet121	4.26	2.09	5.05	4.40	1.40	**0.62**	2.21(2.33)	2.09(1.00)	2.26(2.52)	3.31(0.94)	1.40(1.00)	**0.59(1.02)**
CIFAR-100	ResNet50	17.17	6.57	15.28	7.86	5.64	**1.36**	3.71(2.16)	3.66(1.13)	2.32(1.80)	4.10(1.13)	2.97(1.17)	**1.36(1.00)**
	ResNet110	19.44	7.70	19.11	11.18	7.08	**1.40**	6.11(2.28)	4.55(1.18)	4.88(2.32)	8.58(1.09)	3.85(1.20)	**1.40(1.00)**
	WideResNet	14.83	4.27	13.12	5.10	2.25	**1.95**	3.23(2.12)	2.85(1.08)	4.23(1.91)	5.10(1.00)	2.25(1.00)	**1.95(1.00)**
	DenseNet121	19.82	5.14	19.16	12.81	2.58	**1.73**	3.62(2.27)	2.58(1.09)	3.11(2.13)	8.95(1.19)	1.80(1.10)	**1.73(1.00)**
TinyImageNet	ResNet50	7.81	3.42	8.49	9.12	2.86	**2.61**	3.73(1.45)	2.98(0.93)	4.25(1.36)	4.66(0.78)	2.48(1.05)	**2.29(0.96)**
	ResNet110	8.11	3.74	7.40	9.36	1.88	**1.85**	1.93(1.20)	2.83(0.91)	1.95(1.20)	4.51(0.83)	1.88(1.00)	**1.85(1.00)**
ImageNet	ResNet50	2.93	3.91	9.30	10.05	16.77	**1.87**	**1.50(0.88)**	3.59(0.92)	4.22(1.34)	4.53(0.82)	2.62(0.74)	1.87(1.00)
	ResNet110	1.28	3.98	1.83	4.02	18.66	**1.17**	1.28(1.00)	2.87(0.90)	1.83(1.00)	2.76(0.90)	2.51(0.70)	**1.17(1.00)**
	DenseNet121	1.82	2.94	**1.22**	5.30	19.19	1.50	1.82(1.00)	2.21(0.90)	**1.22(1.00)**	1.42(0.90)	2.24(0.70)	1.50(1.00)
20Newsgroup	CNN	18.57	13.52	15.23	4.36	8.86	**2.62**	4.08(3.78)	3.13(2.33)	6.45(2.21)	2.62(1.12)	**2.13(1.58)**	2.46(1.10)
	BERT	8.47	5.91	8.30	6.01	8.63	**3.96**	4.46(1.44)	4.40(1.24)	4.60(1.46)	5.69(1.14)	3.91(0.80)	**3.73(1.04)**

Table 1: Test ECE_{EM} (%) averaged over 5 runs. Bold marks the lowest in pre and post temperature scaling groups separately. Optimal temperature, given in brackets, is cross-validated on ECE_{EM}.

Test Set ECE (all results)

Dataset	Model	Pre Temperature scaling						Post Temperature scaling					
		CE	Brier	MMCE	LS	FLSD-53	AdaFocal	CE	Brier	MMCE	LS	FLSD-53	AdaFocal
CIFAR-10	ResNet50	4.24	1.78	4.52	3.86	1.63	**0.66**	2.11(2.52)	1.24(1.11)	2.12(2.65)	2.97(0.92)	1.42(1.08)	**0.44(1.06)**
	ResNet110	4.39	2.63	5.16	4.44	1.90	**0.71**	2.27(2.74)	1.75(1.21)	2.53(2.83)	4.44(1.00)	1.25(1.20)	**0.73(1.02)**
	WideResNet	3.42	1.72	3.31	4.26	1.82	**0.64**	1.87(2.16)	1.72(1.00)	1.6(2.22)	2.44(0.81)	1.57(0.94)	**0.44(1.06)**
	DenseNet121	4.26	2.09	5.05	4.40	1.40	**0.62**	2.21(2.33)	2.09(1.00)	2.26(2.52)	3.31(0.94)	1.40(1.00)	**0.59(1.02)**
CIFAR-100	ResNet50	17.17	6.57	15.28	7.86	5.64	**1.36**	3.71(2.16)	3.66(1.13)	2.32(1.80)	4.10(1.13)	2.97(1.17)	**1.36(1.00)**
	ResNet110	19.44	7.70	19.11	11.18	7.08	**1.40**	6.11(2.28)	4.55(1.18)	4.88(2.32)	8.58(1.09)	3.85(1.20)	**1.40(1.00)**
	WideResNet	14.83	4.27	13.12	5.10	2.25	**1.95**	3.23(2.12)	2.85(1.08)	4.23(1.91)	5.10(1.00)	2.25(1.00)	**1.95(1.00)**
	DenseNet121	19.82	5.14	19.16	12.81	2.58	**1.73**	3.62(2.27)	2.58(1.09)	3.11(2.13)	8.95(1.19)	1.80(1.10)	**1.73(1.00)**
TinyImageNet	ResNet50	7.81	3.42	8.49	9.12	2.86	**2.61**	3.73(1.45)	2.98(0.93)	4.25(1.36)	4.66(0.78)	2.48(1.05)	**2.29(0.96)**
	ResNet110	8.11	3.74	7.40	9.36	1.88	**1.85**	1.93(1.20)	2.83(0.91)	1.95(1.20)	4.51(0.83)	1.88(1.00)	**1.85(1.00)**
ImageNet	ResNet50	2.93	3.91	9.30	10.05	16.77	**1.87**	**1.50(0.88)**	3.59(0.92)	4.22(1.34)	4.53(0.82)	2.62(0.74)	1.87(1.00)
	ResNet110	1.28	3.98	1.83	4.02	18.66	**1.17**	1.28(1.00)	2.87(0.90)	1.83(1.00)	2.76(0.90)	2.51(0.70)	**1.17(1.00)**
	DenseNet121	1.82	2.94	**1.22**	5.30	19.19	1.50	1.82(1.00)	2.21(0.90)	**1.22(1.00)**	1.42(0.90)	2.24(0.70)	1.50(1.00)
20Newsgroup	CNN	18.57	13.52	15.23	4.36	8.86	**2.62**	4.08(3.78)	3.13(2.33)	6.45(2.21)	2.62(1.12)	**2.13(1.58)**	2.46(1.10)
	BERT	8.47	5.91	8.30	6.01	8.63	**3.96**	4.46(1.44)	4.40(1.24)	4.60(1.46)	5.69(1.14)	3.91(0.80)	**3.73(1.04)**

Table 1: Test ECE_{EM} (%) averaged over 5 runs. Bold marks the lowest in pre and post temperature scaling groups separately. Optimal temperature, given in brackets, is cross-validated on ECE_{EM}.

Among “calibration-during-training” methods, AdaFocal achieves the best result in 14/15 cases.

Test Set ECE (all results)

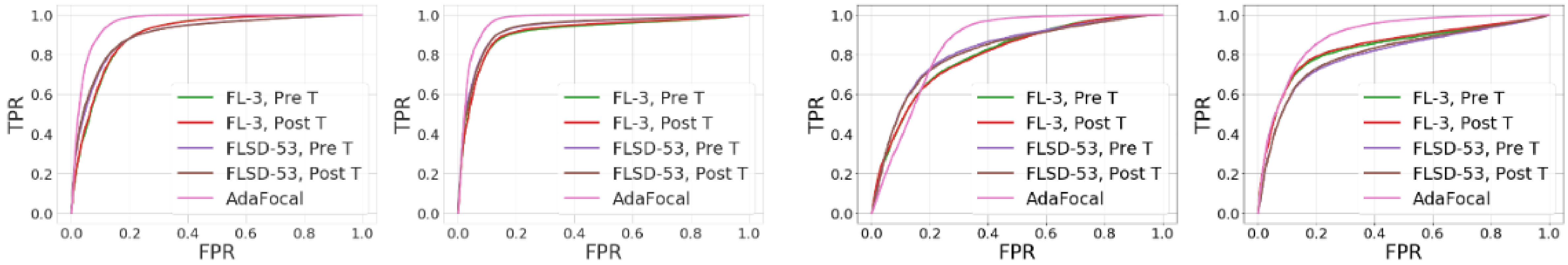
Dataset	Model	Pre Temperature scaling						Post Temperature scaling					
		CE	Brier	MMCE	LS	FLSD-53	AdaFocal	CE	Brier	MMCE	LS	FLSD-53	AdaFocal
CIFAR-10	ResNet50	4.24	1.78	4.52	3.86	1.63	**0.66**	2.11(2.52)	1.24(1.11)	2.12(2.65)	2.97(0.92)	1.42(1.08)	**0.44(1.06)**
	ResNet110	4.39	2.63	5.16	4.44	1.90	**0.71**	2.27(2.74)	1.75(1.21)	2.53(2.83)	4.44(1.00)	1.25(1.20)	**0.73(1.02)**
	WideResNet	3.42	1.72	3.31	4.26	1.82	**0.64**	1.87(2.16)	1.72(1.00)	1.6(2.22)	2.44(0.81)	1.57(0.94)	**0.44(1.06)**
	DenseNet121	4.26	2.09	5.05	4.40	1.40	**0.62**	2.21(2.33)	2.09(1.00)	2.26(2.52)	3.31(0.94)	1.40(1.00)	**0.59(1.02)**
CIFAR-100	ResNet50	17.17	6.57	15.28	7.86	5.64	**1.36**	3.71(2.16)	3.66(1.13)	2.32(1.80)	4.10(1.13)	2.97(1.17)	**1.36(1.00)**
	ResNet110	19.44	7.70	19.11	11.18	7.08	**1.40**	6.11(2.28)	4.55(1.18)	4.88(2.32)	8.58(1.09)	3.85(1.20)	**1.40(1.00)**
	WideResNet	14.83	4.27	13.12	5.10	2.25	**1.95**	3.23(2.12)	2.85(1.08)	4.23(1.91)	5.10(1.00)	2.25(1.00)	**1.95(1.00)**
	DenseNet121	19.82	5.14	19.16	12.81	2.58	**1.73**	3.62(2.27)	2.58(1.09)	3.11(2.13)	8.95(1.19)	1.80(1.10)	**1.73(1.00)**
TinyImageNet	ResNet50	7.81	3.42	8.49	9.12	2.86	**2.61**	3.73(1.45)	2.98(0.93)	4.25(1.36)	4.66(0.78)	2.48(1.05)	**2.29(0.96)**
	ResNet110	8.11	3.74	7.40	9.36	1.88	**1.85**	1.93(1.20)	2.83(0.91)	1.95(1.20)	4.51(0.83)	1.88(1.00)	**1.85(1.00)**
ImageNet	ResNet50	2.93	3.91	9.30	10.05	16.77	**1.87**	**1.50(0.88)**	3.59(0.92)	4.22(1.34)	4.53(0.82)	2.62(0.74)	1.87(1.00)
	ResNet110	1.28	3.98	1.83	4.02	18.66	**1.17**	1.28(1.00)	2.87(0.90)	1.83(1.00)	2.76(0.90)	2.51(0.70)	**1.17(1.00)**
	DenseNet121	1.82	2.94	**1.22**	5.30	19.19	1.50	1.82(1.00)	2.21(0.90)	**1.22(1.00)**	1.42(0.90)	2.24(0.70)	1.50(1.00)
20Newsgroup	CNN	18.57	13.52	15.23	4.36	8.86	**2.62**	4.08(3.78)	3.13(2.33)	6.45(2.21)	2.62(1.12)	**2.13(1.58)**	2.46(1.10)
	BERT	8.47	5.91	8.30	6.01	8.63	**3.96**	4.46(1.44)	4.40(1.24)	4.60(1.46)	5.69(1.14)	3.91(0.80)	**3.73(1.04)**

Table 1: Test ECE_{EM} (%) averaged over 5 runs. Bold marks the lowest in pre and post temperature scaling groups separately. Optimal temperature, given in brackets, is cross-validated on ECE_{EM}.

AdaFocal produces inherently calibrated models that benefit further from post-hoc calibration such as temperature scaling, outperforming in 12/15 cases.

Out-of-Distribution Detection Task

Out-of-Distribution Detection Task

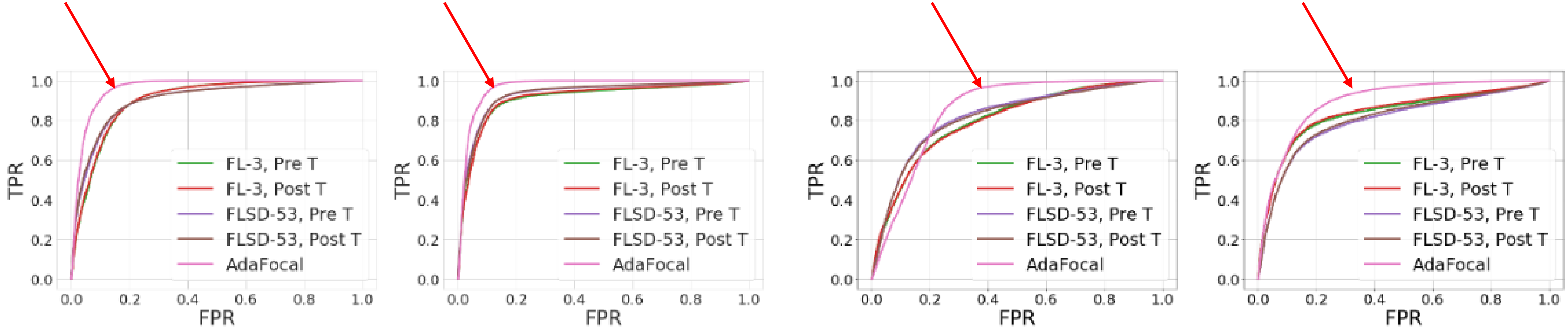


(a) SVHN: ResNet-110, WideResNet

(b) CIFAR-10-C: ResNet-110, WideResNet

Figure 6: ROC for ResNet-110 and Wide-ResNet-26-10 trained on in-distribution CIFAR-10 and tested on out-of-distribution (a) SVHN and (b) CIFAR-10-C. Pre/Post T refers to pre and post temperature scaling.

Out-of-Distribution Detection Task



(a) SVHN: ResNet-110, WideResNet

(b) CIFAR-10-C: ResNet-110, WideResNet

Figure 6: ROC for ResNet-110 and Wide-ResNet-26-10 trained on in-distribution CIFAR-10 and tested on out-of-distribution (a) SVHN and (b) CIFAR-10-C. Pre/Post T refers to pre and post temperature scaling.

Thank you!