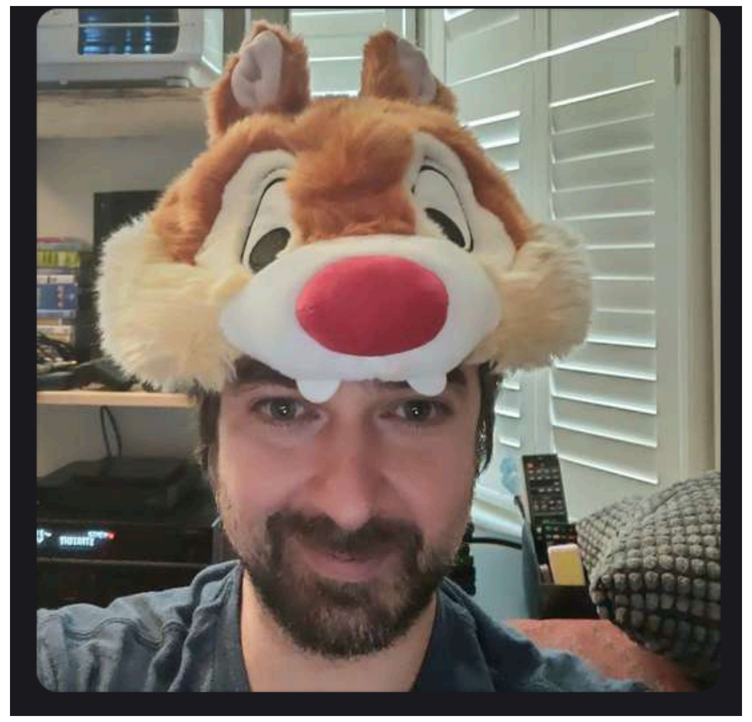
Calibration of Deep Neural Networks and Beyond...

Puneet K. Dokania
(University of Oxford and Five AI Oxford)

ICCV Paris Tutorial on 3rd Oct 2023 (The Many Faces of Reliability of Deep Learning for Real-World Deployment)

Five AI Oxford Team



Nick Lord

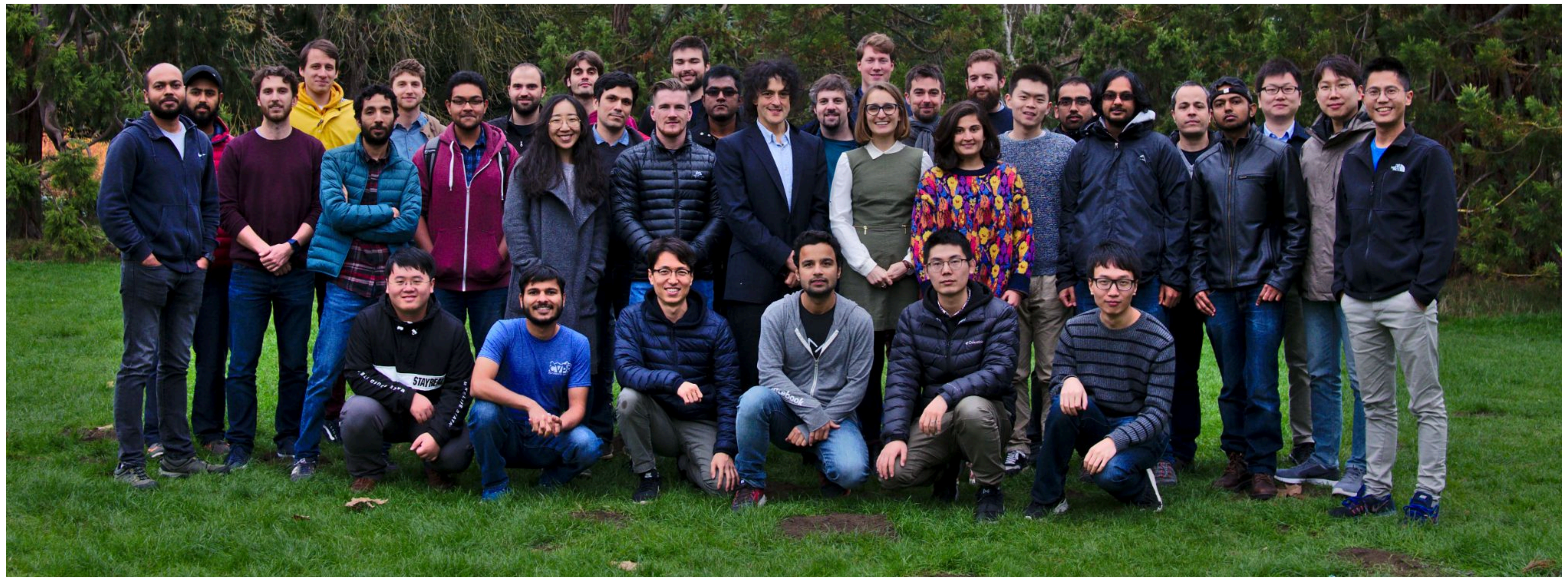
Kemal Oksuz

Tom Joy

Selim Kuzucu

Torr Vision Group (Oxford Uni)

- Prof. Philip Torr (Lead)
- PhD and Postdocs at TVG

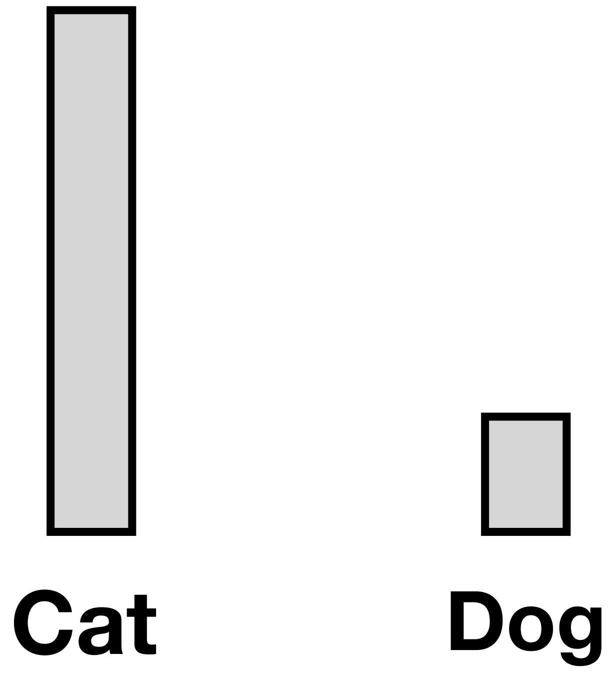


(Old photo)

Background

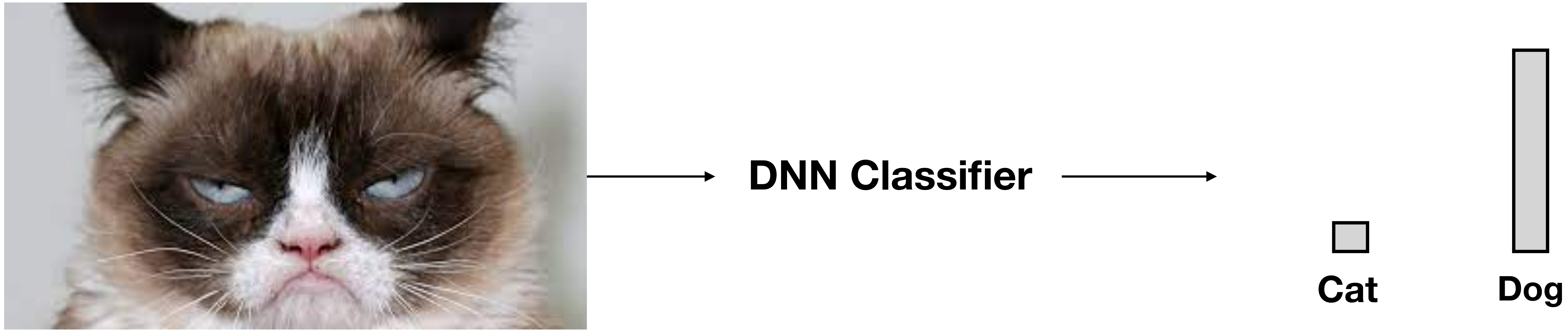
Neural Networks are mostly highly accurate and confident …

DNN Classifier



Background

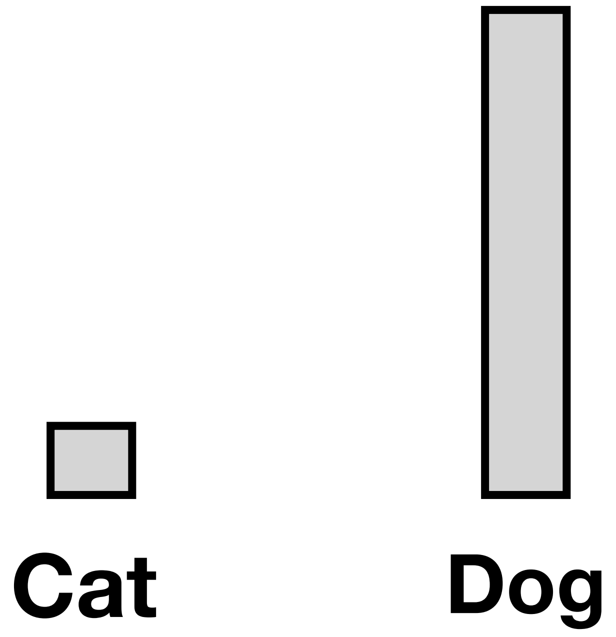
… and very confident even when wrong (miscalibration)



Background

… and very confident on Out-of-distribution data (unaware of unknowns)

DNN Classifier

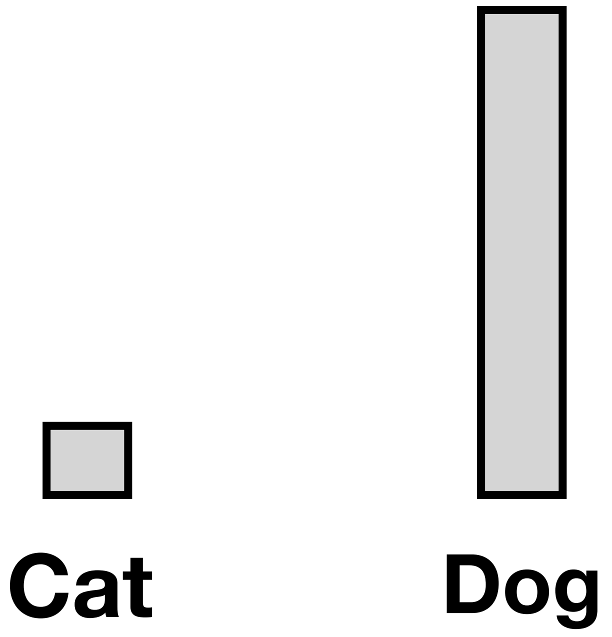


(ideally, we would like them to be uncertain on these samples)

Background

… and often wrong under covariate shift

DNN Classifier



Background

Why these problems?

Background

Why these problems?

- Why confident even when wrong?
- Why not uncertain on OOD data?
- Why can't they handle covariate shift?

- Overparameterization and high capacity?
- Poor regularisation?
- Spurious features?
- Models not expressive enough?
- Less data?

Background

Why these problems?

- Why confident even when wrong?
- Why not uncertain on OOD data?
- Why can't they handle covariate shift?

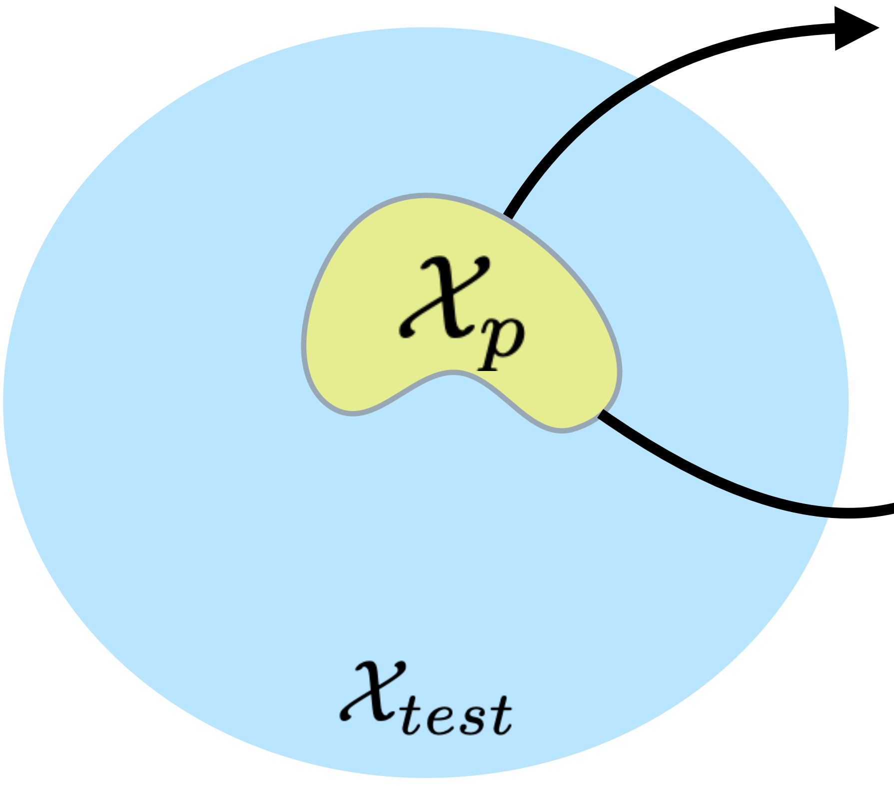
- Overparameterization and high capacity?
- Poor regularisation?
- Spurious features?
- Models not expressive enough?
- Less data?

Perhaps because we didn't ask them to be robust?

(Our design choices are mainly focused towards minimizing generalisation gap on a relatively small dataset under controlled environment on metrics focused towards accuracy — far from the real-world situations we would like answers for)

Set-up

First definition (Probability Calibration — The real one)

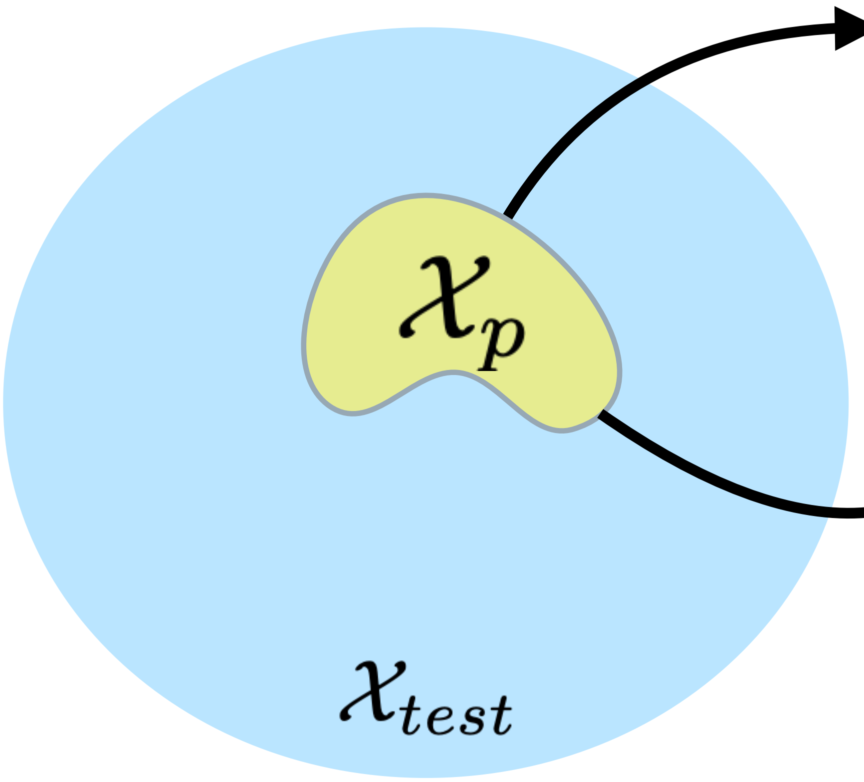


$$P_\theta(\mathbf{y}|\mathbf{x}) = p, \forall \mathbf{x} \in \mathcal{X}_p$$

$$\hat{P}(\mathbf{y}|\mathcal{X}_p)$$

Set-up

First definition (Probability Calibration — The real one)



$$P_\theta(\mathbf{y}|\mathbf{x}) = p, \forall \mathbf{x} \in \mathcal{X}_p$$

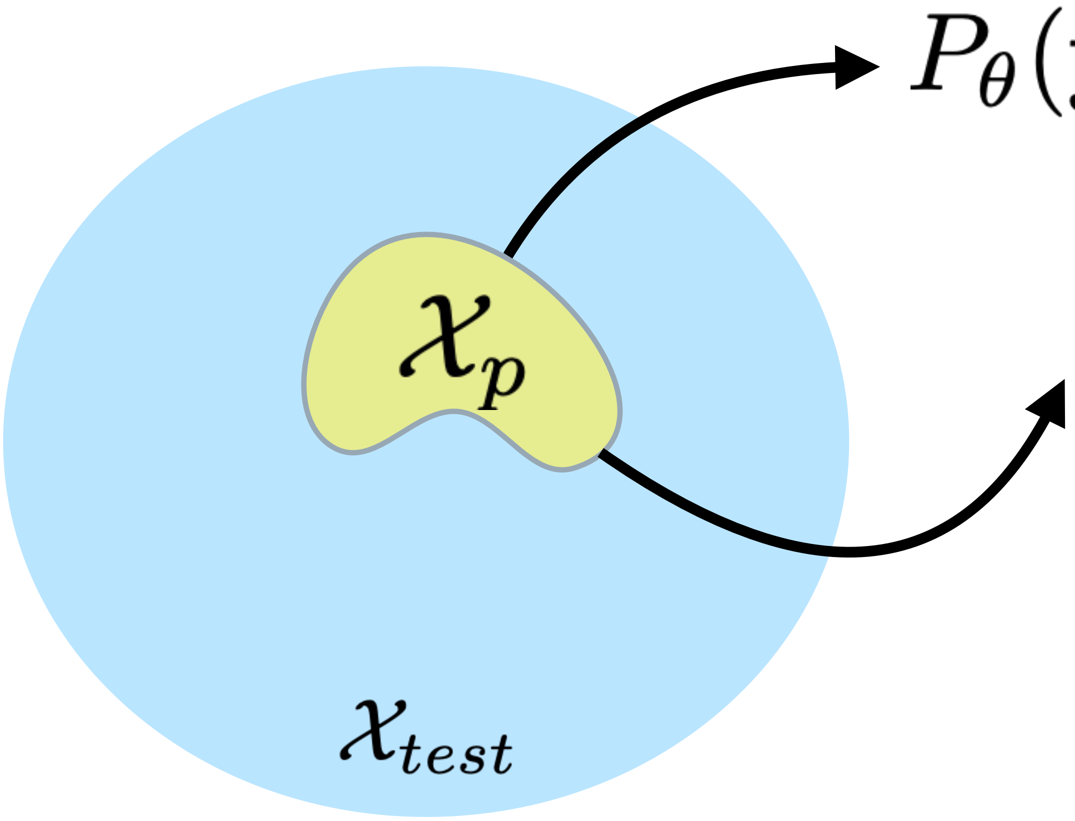
$$\hat{P}(\mathbf{y}|\mathcal{X}_p)$$

$$\hat{P}(\mathbf{y}|\mathcal{X}_p) = P_\theta(\mathbf{y}|\mathcal{X}_p) = p$$

$$\forall p$$

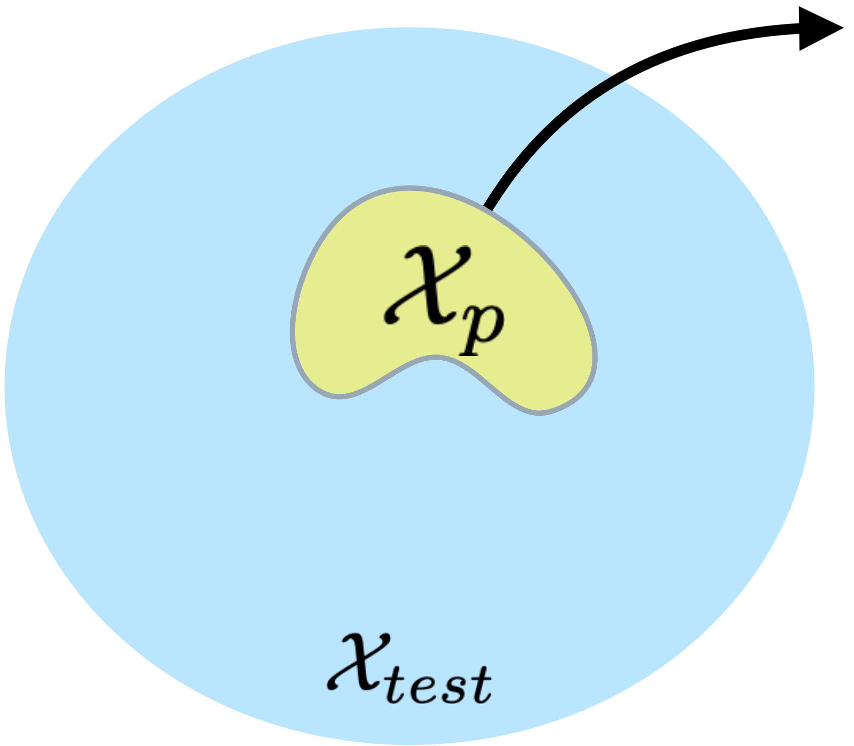
Set-up

First definition (Probability Calibration — The real one)



$$\left.\begin{array}{l} P_\theta(\mathbf{y}|\mathbf{x}) = p, \forall \mathbf{x} \in \mathcal{X}_p \\ \\ \hat{P}(\mathbf{y}|\mathcal{X}_p) \end{array}\right\} \quad \begin{array}{c} \hat{P}(\mathbf{y}|\mathcal{X}_p) = P_\theta(\mathbf{y}|\mathcal{X}_p) = p \\ \forall p \end{array}$$

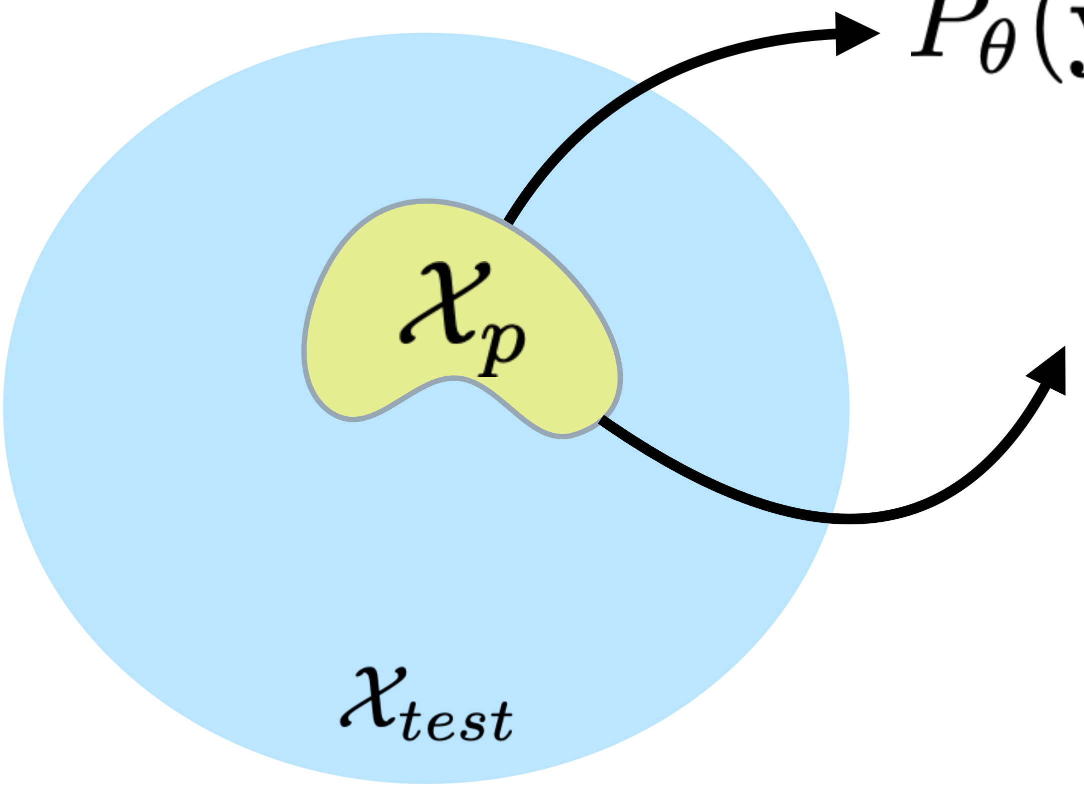
Second definition (Confidence Calibration — Mostly used in DNNs)



$$\max P_\theta(\mathbf{y}|\mathbf{x}) = p, \forall \mathbf{x} \in \mathcal{X}_p$$

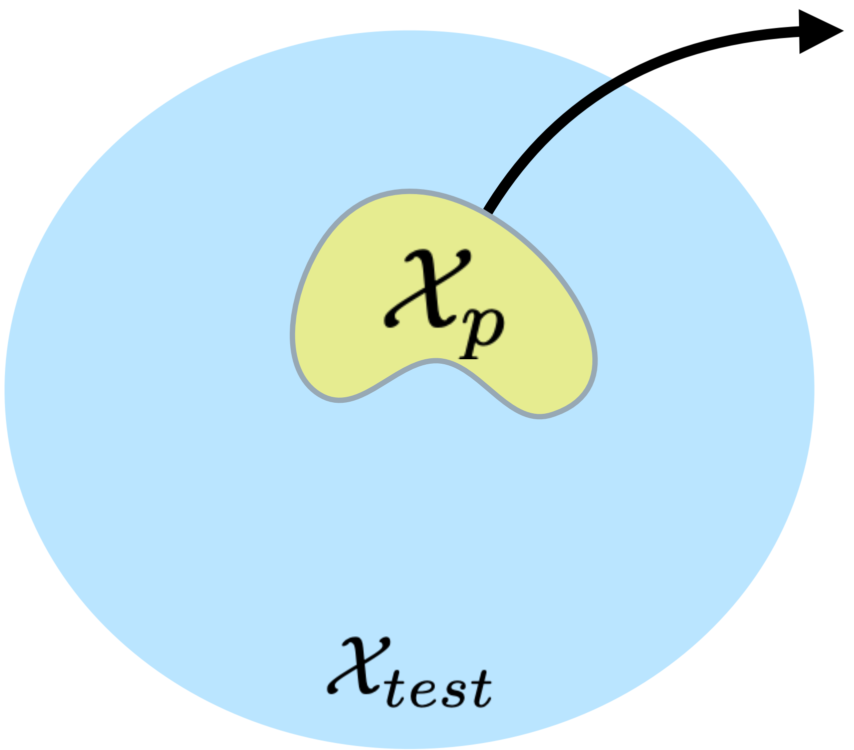
Set-up

First definition (Probability Calibration — The real one)



$$\left.\begin{array}{l} P_\theta(\mathbf{y}|\mathbf{x}) = p, \forall \mathbf{x} \in \mathcal{X}_p \\ \hat{P}(\mathbf{y}|\mathcal{X}_p) \end{array}\right\} \quad \begin{array}{c} \hat{P}(\mathbf{y}|\mathcal{X}_p) = P_\theta(\mathbf{y}|\mathcal{X}_p) = p \\ \forall p \end{array}$$

Second definition (Confidence Calibration — Mostly used in DNNs)

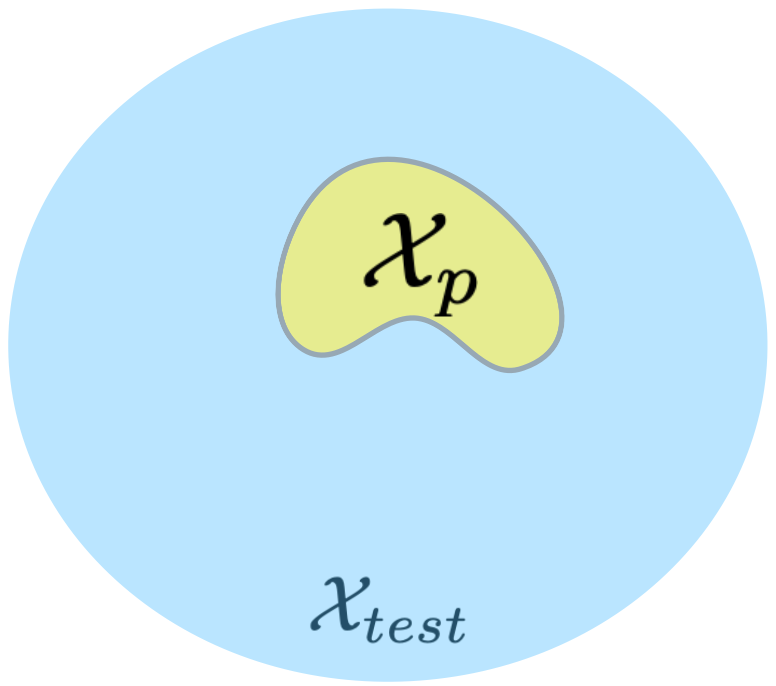


$$\max P_\theta(\mathbf{y}|\mathbf{x}) = p, \forall \mathbf{x} \in \mathcal{X}_p$$

$$\underbrace{\bar{P}\left(y = \text{argmax}\ P_\theta(\mathbf{y}|\mathcal{X}_p)|\mathcal{X}_p\right)}_{accuracy} = \underbrace{\max P_\theta(\mathbf{y}|\mathcal{X}_p)}_{confidence} = p, \forall p \in [0, 1]$$

Set-up

Examples



$$\mathcal{X}_{test} = \{70 \text{ cats}, 20 \text{ dogs}, 10 \text{ birds}\}$$

- **Case 1:**

$$P_\theta(\mathbf{y}|\mathbf{x}) = \begin{pmatrix} 0.7 \\ 0.2 \\ 0.1 \end{pmatrix}, \forall \mathbf{x} \in \mathcal{X}_{test} \implies \mathcal{X}_{test} = \mathcal{X}_p$$

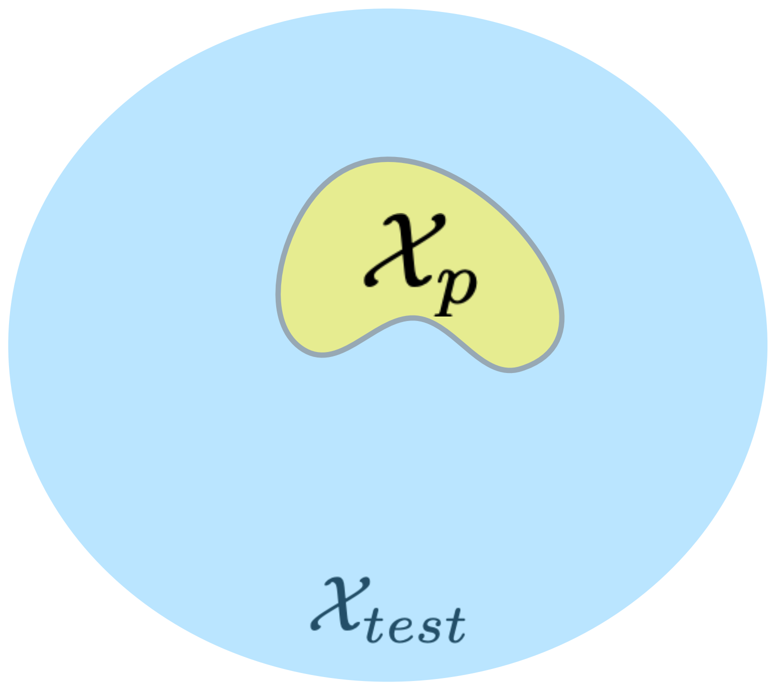
$$\underbrace{\bar{P}\Big(y = \text{argmax } P_\theta(\mathbf{y}|\mathcal{X}_p)|\mathcal{X}_p\Big)}_{accuracy} = \underbrace{\max P_\theta(\mathbf{y}|\mathcal{X}_p)}_{confidence} = 0.7$$

$$\hat{P}(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p) = \begin{pmatrix} 0.7 \\ 0.2 \\ 0.1 \end{pmatrix} = P_\theta(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p)$$

- **Confidence Calibrated**
- **Probability Calibrated**

Set-up

Examples



$$\mathcal{X}_{test} = \{70 \text{ cats}, 20 \text{ dogs}, 10 \text{ birds}\}$$

- **Case 1:**

$$P_\theta(\mathbf{y}|\mathbf{x}) = \begin{pmatrix} 0.7 \\ 0.2 \\ 0.1 \end{pmatrix}, \forall \mathbf{x} \in \mathcal{X}_{test} \implies \mathcal{X}_{test} = \mathcal{X}_p$$

$$\underbrace{\bar{P}\Big(y = \text{argmax}\, P_\theta(\mathbf{y}|\mathcal{X}_p)|\mathcal{X}_p\Big)}_{accuracy} = \underbrace{\max P_\theta(\mathbf{y}|\mathcal{X}_p)}_{confidence} = 0.7$$

$$\hat{P}(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p) = \begin{pmatrix} 0.7 \\ 0.2 \\ 0.1 \end{pmatrix} = P_\theta(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p)$$

- **Confidence Calibrated**
- **Probability Calibrated**

- **Case 2:**

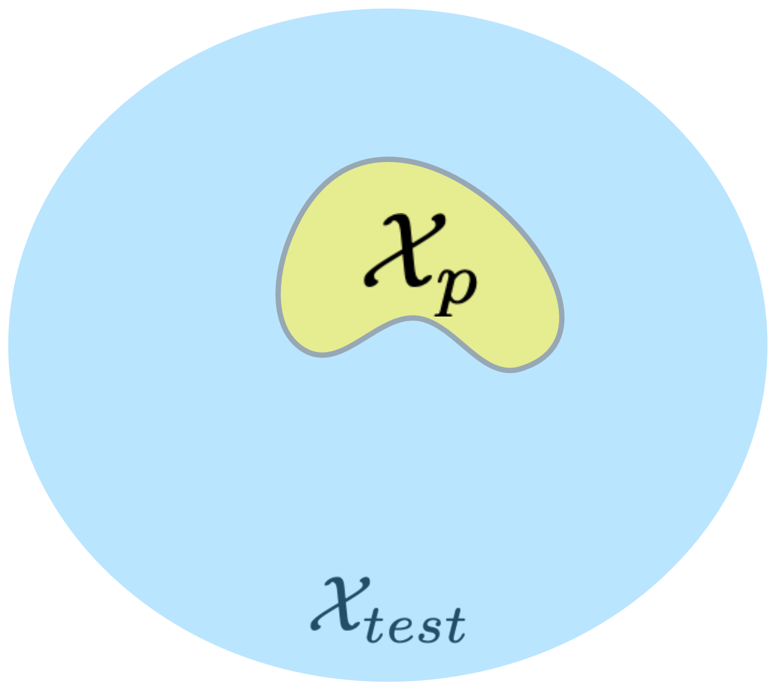
$$P_\theta(\mathbf{y}|\mathbf{x}) = \begin{pmatrix} 0.7 \\ 0.18 \\ 0.12 \end{pmatrix}, \forall \mathbf{x} \in \mathcal{X}_{test} \implies \mathcal{X}_{test} = \mathcal{X}_p$$

$$\hat{P}(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p) \neq P_\theta(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p)$$

- **Confidence Calibrated**
- **Probability Uncalibrated**

Set-up

Examples



$$\mathcal{X}_{test} = \{70 \text{ cats}, 20 \text{ dogs}, 10 \text{ birds}\}$$

- **Case 1:**

$$P_\theta(\mathbf{y}|\mathbf{x}) = \begin{pmatrix} 0.7 \\ 0.2 \\ 0.1 \end{pmatrix}, \forall \mathbf{x} \in \mathcal{X}_{test} \implies \mathcal{X}_{test} = \mathcal{X}_p$$

$$\underbrace{\bar{P}\Big(y = \text{argmax } P_\theta(\mathbf{y}|\mathcal{X}_p)|\mathcal{X}_p\Big)}_{accuracy} = \underbrace{\max P_\theta(\mathbf{y}|\mathcal{X}_p)}_{confidence} = 0.7$$

- **Confidence Calibrated**
- **Probability Calibrated**

$$\hat{P}(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p) = \begin{pmatrix} 0.7 \\ 0.2 \\ 0.1 \end{pmatrix} = P_\theta(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p)$$

- **Case 2:**

$$P_\theta(\mathbf{y}|\mathbf{x}) = \begin{pmatrix} 0.7 \\ 0.18 \\ 0.12 \end{pmatrix}, \forall \mathbf{x} \in \mathcal{X}_{test} \implies \mathcal{X}_{test} = \mathcal{X}_p$$

- **Confidence Calibrated**
- **Probability Uncalibrated**

$$\hat{P}(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p) \neq P_\theta(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p)$$

$$P_\theta(\mathbf{y}|\mathbf{x} \in \mathcal{X}_p) = \begin{pmatrix} 0.7 \\ \epsilon_+ \\ 0.3 - \epsilon_+ \end{pmatrix}$$

- **Probability Calibrated only for \eps_+ = 0.2**
- **Confidence Calibrated throughout**
- **70% Accurate throughout**

Quantifying Miscalibration

Confidence calibration — ECE, AdaECE, Reliability Histogram

$$\underbrace{P\Big(\hat{y} = \text{argmax}\ P_\theta(\mathbf{y}|\mathcal{X}_p)|\mathcal{X}_p\Big)}_{accuracy} = \underbrace{\max P_\theta(\mathbf{y}|\mathcal{X}_p)}_{confidence}, \forall p$$

- Expected mismatch between accuracy and confidence [Naeini et al., AAAI15]
  - Discretize the space (binning)
  - Compute the mismatch/bin

$$\text{ECE} = \sum_{i=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|$$

Quantifying Miscalibration

Confidence calibration — ECE, AdaECE, Reliability Histogram

$$\underbrace{P\Big(\hat{y} = \operatorname{argmax} P_\theta(\mathbf{y}|\mathcal{X}_p)|\mathcal{X}_p\Big)}_{accuracy} = \underbrace{\max P_\theta(\mathbf{y}|\mathcal{X}_p)}_{confidence}, \forall p$$

- Expected mismatch between accuracy and confidence [Naeini et al., AAAI15]
  - Discretize the space (binning)
  - Compute the mismatch/bin

$$\text{ECE} = \sum_{i=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|$$

- Other variants
  - Maximum Calibration Error [Naeini et al., AAAI15]
  - Class-wise ECE (stronger defn than ECE) [Kull et al., NeurIPS19]
  - Adaptive ECE [Mukhoti et al., NeurIPS20]

Calibration - Why?

- Uncalibrated models can be wrong with high confidence
  - Autonomous driving
  - Medical Imaging etc.

- Are NNs uncalibrated?
  - Many studies have shown that they are, but why?
    - No single answer unfortunately

Calibration — How?

Post-hoc (Temperature Scaling)

- Temperature scaling — on a val set, find a positive temperature hyper-parameters to ensure decrease in ECE, while not modifying their accuracies [Guo et al., ICML17]

$$\sigma_T(s_i) = \frac{\exp(\frac{s_i}{T})}{\sum_j \exp(\frac{s_j}{T})}$$

- Works well as DNNs are overconfident but this approach effectively makes a model under-confident as T generally is >1.

Calibration — How?

Post-hoc (Adaptive Temperature Scaling [Joy et al., AAAI 23])

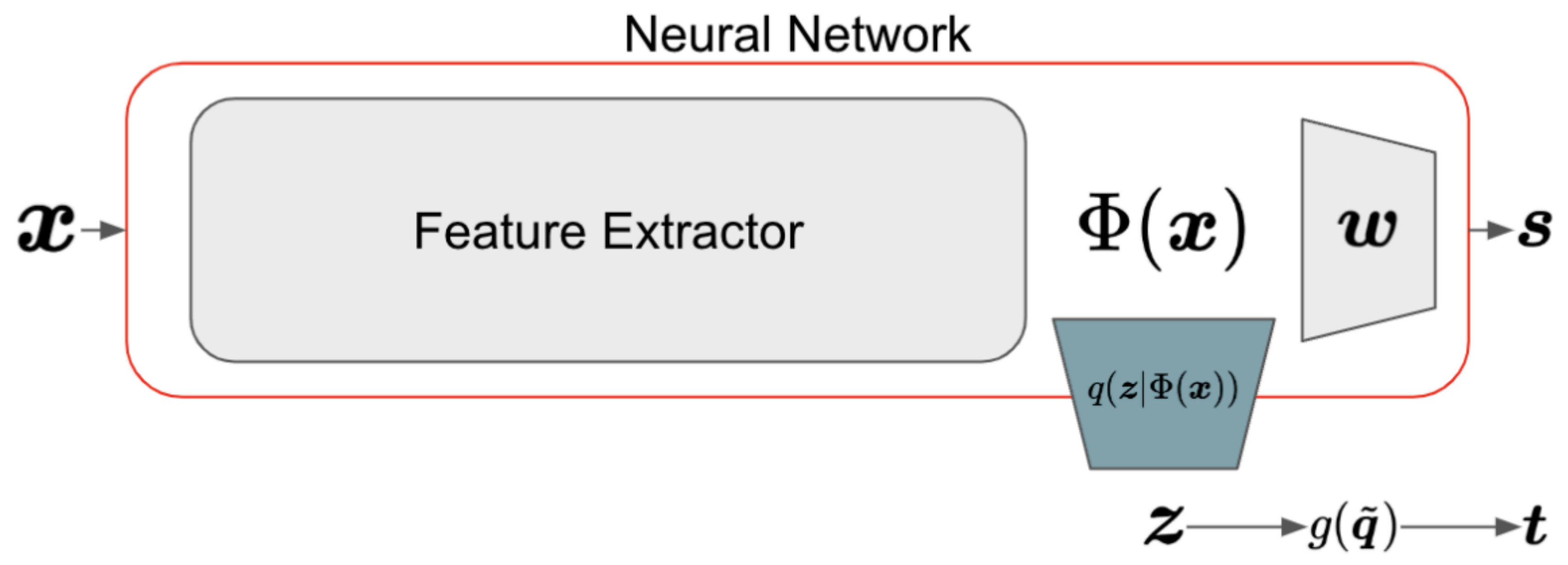
- Temperature scaling — Same T for all the samples
- However, different samples contribute differently to miscalibration

Calibration — How?

Post-hoc (Adaptive Temperature Scaling [Joy et al., AAAI 23])

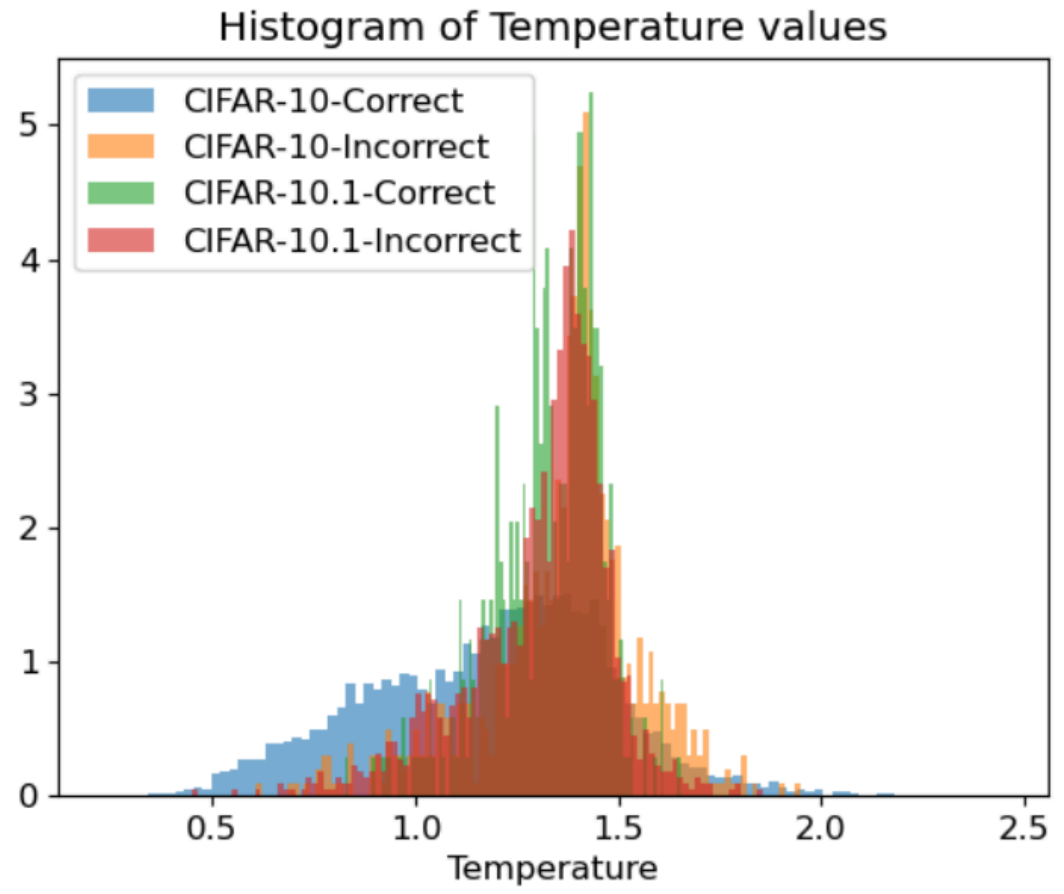
- Temperature scaling — Same T for all the samples

- However, different samples contribute differently to miscalibration



$$\mathcal{L}(\mathbf{x}, y) = \mathbb{ELBO}[\Phi(\mathbf{x})] + \log \mathrm{Cat}(y \mid \mathrm{softmax}(\mathbf{s}/g_\theta(\tilde{\boldsymbol{q}}))$$

- VAE objective over feature space
  - Assuming feature space contains the necessary info
- Loglikelihood to ensure the error-behaviour on the val set remains the same.



Calibration — How?

During training (Focal loss, MMCE, Brier Score, Label Smoothing)

Calibration — How?

During training (Focal loss, MMCE, Brier Score, Label Smoothing)

Dataset	Model	Cross-Entropy		Brier Loss		MMCE		LS-0.05		FL-3 (Ours)		FLSD-53 (Ours)	
		Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T
CIFAR-100	ResNet-50	17.52	3.42(2.1)	6.52	3.64(1.1)	15.32	2.38(1.8)	7.81	4.01(1.1)	5.13	**1.97(1.1)**	**4.5**	2.0(1.1)
	ResNet-110	19.05	4.43(2.3)	**7.88**	4.65(1.2)	19.14	**3.86(2.3)**	11.02	5.89(1.1)	8.64	3.95(1.2)	8.56	4.12(1.2)
	Wide-ResNet-26-10	15.33	2.88(2.2)	4.31	2.7(1.1)	13.17	4.37(1.9)	4.84	4.84(1)	**2.13**	2.13(1)	3.03	**1.64(1.1)**
	DenseNet-121	20.98	4.27(2.3)	5.17	2.29(1.1)	19.13	3.06(2.1)	12.89	7.52(1.2)	4.15	**1.25(1.1)**	**3.73**	1.31(1.1)
CIFAR-10	ResNet-50	4.35	1.35(2.5)	1.82	1.08(1.1)	4.56	1.19(2.6)	2.96	1.67(0.9)	**1.48**	1.42(1.1)	1.55	**0.95(1.1)**
	ResNet-110	4.41	1.09(2.8)	2.56	1.25(1.2)	5.08	1.42(2.8)	2.09	2.09(1)	**1.55**	**1.02(1.1)**	1.87	1.07(1.1)
	Wide-ResNet-26-10	3.23	0.92(2.2)	**1.25**	1.25(1)	3.29	0.86(2.2)	4.26	1.84(0.8)	1.69	0.97(0.9)	1.56	**0.84(0.9)**
	DenseNet-121	4.52	1.31(2.4)	1.53	1.53(1)	5.1	1.61(2.5)	1.88	1.82(0.9)	1.32	1.26(0.9)	**1.22**	**1.22(1)**
Tiny-ImageNet	ResNet-50	15.32	5.48(1.4)	4.44	4.13(0.9)	13.01	5.55(1.3)	15.23	6.51(0.7)	1.87	1.87(1)	**1.76**	**1.76(1)**
20 Newsgroups	Global Pooling CNN	17.92	2.39(3.4)	13.58	3.22(2.3)	15.48	6.78(2.2)	**4.79**	2.54(1.1)	8.67	3.51(1.5)	6.92	**2.19(1.5)**
SST Binary	Tree-LSTM	7.37	2.62(1.8)	9.01	2.79(2.5)	5.03	4.02(1.5)	**4.84**	4.11(1.2)	16.05	1.78(0.5)	9.19	**1.83(0.7)**

Table 1: ECE (%) computed for different approaches both pre and post temperature scaling (cross-validating T on ECE). Optimal temperature for each method is indicated in brackets. $T \approx 1$ indicates innately calibrated model.

Is calibration itself enough?

Reliable uncertainty estimation

- Not necessarily
  - It characterises behaviour on in-distribution and perhaps on domain shift datasets
  - What about out-of-distribution?

Is calibration itself enough?

Reliable uncertainty estimation

- Not necessarily
  - It characterises behaviour on in-distribution and perhaps on domain shift datasets
  - What about out-of-distribution?

- Ideally,
  - We would like models to be **accurate and calibrated on in-distribution data**
  - And their **uncertainties to increase** as the input goes *away* from in-distribution
    - Heavy domain shift
    - Out of distribution

Improving Calibration and Uncertainty Estimates

RegMixup [Pinto et al., NeurIPS22]

- Explicitly ask the model to be **uncertain** (under confident) **outside** the data-distribution

$$\min_{\theta} - \log p(y|\mathbf{x} \in \mathcal{X}_I; \theta) - \mathcal{H}_{\bar{y}}(p(.|\mathbf{x} \in \bar{\mathcal{X}}_O; \theta)),$$

Improving Calibration and Uncertainty Estimates

RegMixup [Pinto et al., NeurIPS22]

- Explicitly ask the model to be **uncertain** (under confident) **outside** the data-distribution

$$\min_{\theta} -\log p(y|\mathbf{x} \in \mathcal{X}_I; \theta) - \mathcal{H}_{\bar{y}}(p(.|\mathbf{x} \in \bar{\mathcal{X}}_O; \theta)),$$

- How do we obtain and use OOD data efficiently?

Improving Calibration and Uncertainty Estimates

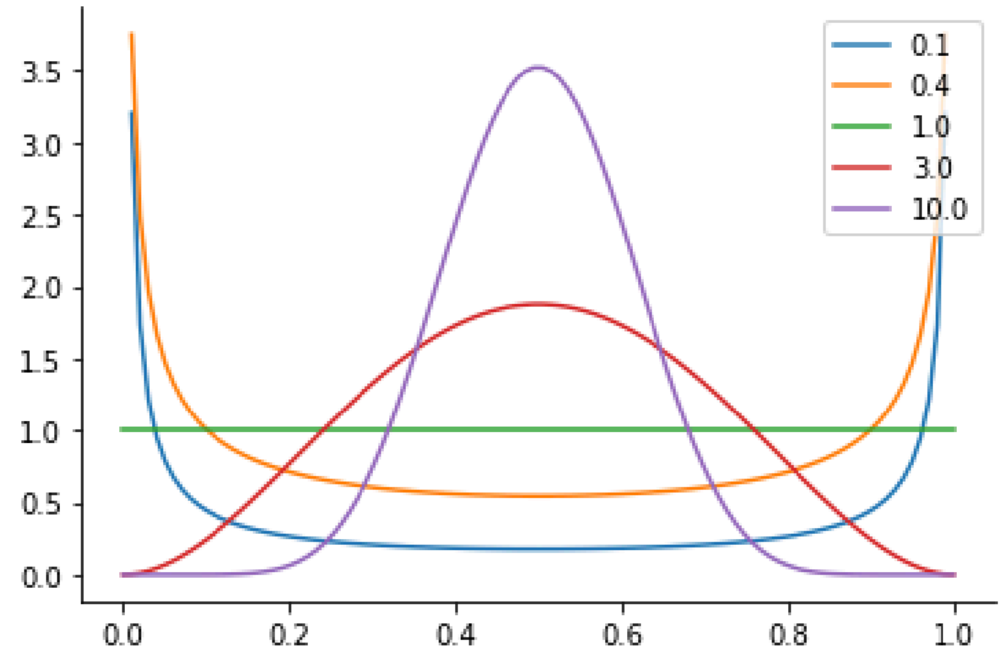
RegMixup [Pinto et al., NeurIPS22]

- Explicitly ask the model to be **uncertain** (under confident) **outside** the data-distribution

$$\min_{\theta} -\log p(y|\mathbf{x} \in \mathcal{X}_I; \theta) - \mathcal{H}_{\bar{y}}(p(.|\mathbf{x} \in \bar{\mathcal{X}}_O; \theta)),$$

- How do we obtain and use OOD data efficiently?
  - Synthesize OOD using Mixup type interpolation

$$\bar{\mathbf{x}} = \lambda_0 \mathbf{x}_i + (1 - \lambda_0)\mathbf{x}_j, \quad if \ y_i \neq y_j.$$



Improving Calibration and Uncertainty Estimates

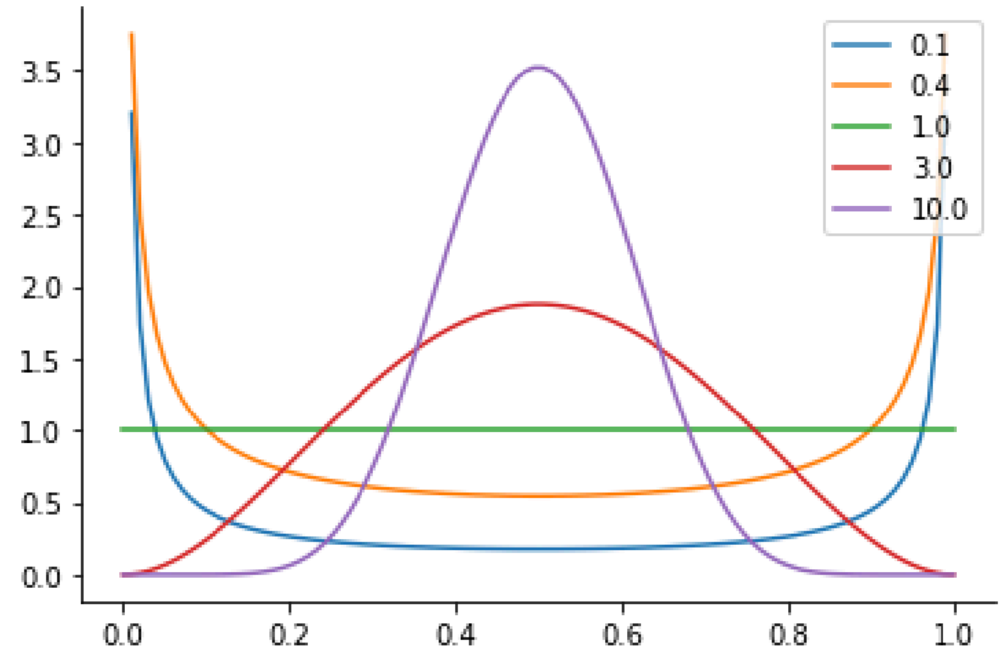
RegMixup [Pinto et al., NeurIPS22]

- Explicitly ask the model to be **uncertain** (under confident) **outside** the data-distribution

$$\min_{\theta} - \log p(y|\mathbf{x} \in \mathcal{X}_I; \theta) - \mathcal{H}_{\bar{y}}(p(.|\mathbf{x} \in \bar{\mathcal{X}}_O; \theta)),$$

- How do we obtain and use OOD data efficiently?
  - Synthesize OOD using Mixup type interpolation

$$\bar{\mathbf{x}} = \lambda_0 \mathbf{x}_i + (1 - \lambda_0)\mathbf{x}_j, \quad if\ y_i \neq y_j.$$



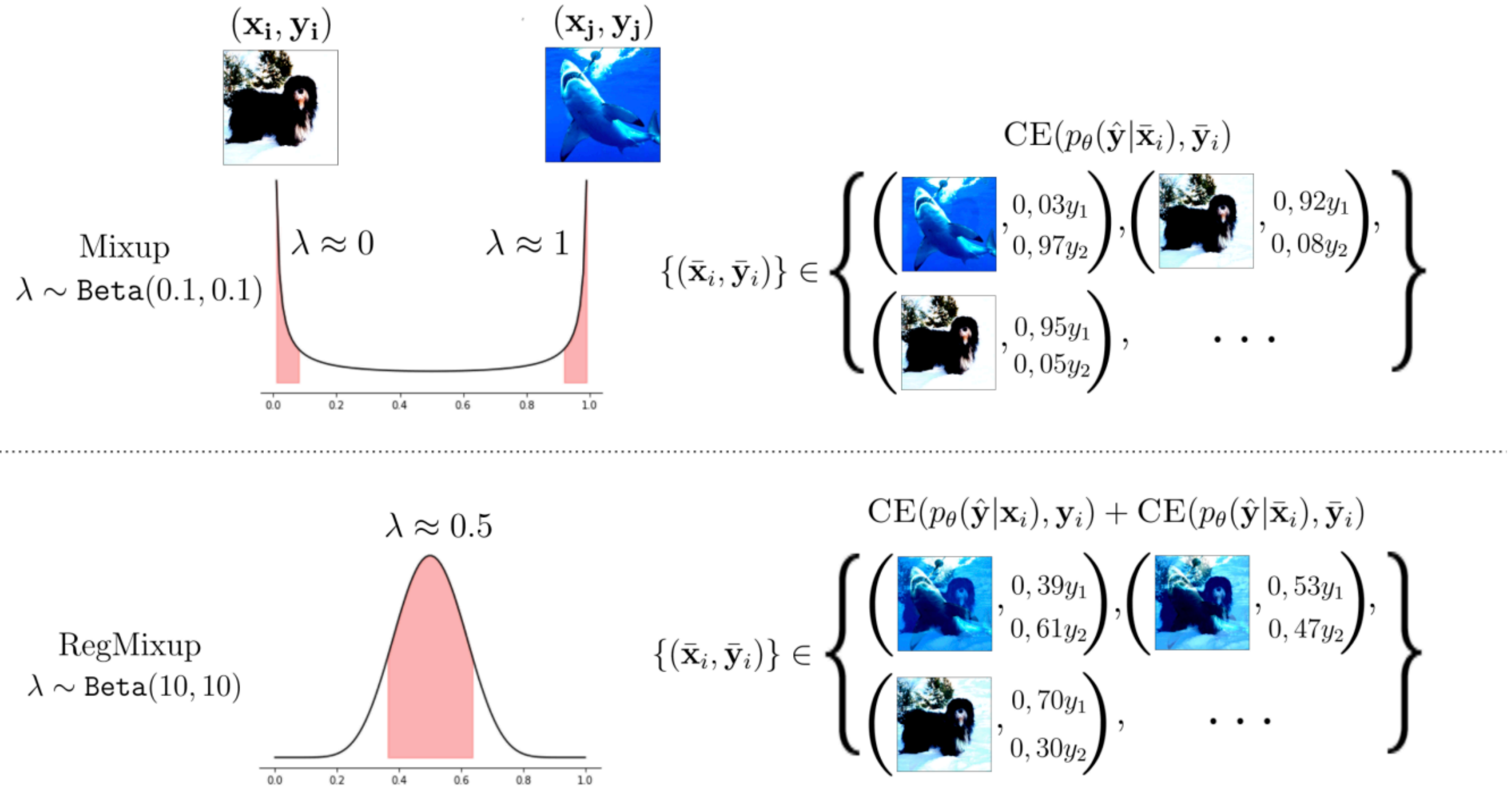
- Final RegMixup Objective

$$\mathrm{CE}(p_\theta(\hat{\mathbf{y}}|\mathbf{x}_i), \mathbf{y}_i) + \eta\ \mathrm{CE}(p_\theta(\hat{\mathbf{y}}|\bar{\mathbf{x}}_i), \bar{\mathbf{y}}_i).$$

(Note, first term is **not** present in standard Mixup — very simple modification yet highly effective)

Improving Calibration and Uncertainty Estimates

RegMixup vs Mixup



Improving Calibration and Uncertainty Estimates

RegMixup vs Mixup

- **Is it doing anything interesting?**
  - WideResNet28-10 trained on CIFAR10
  - 1K pairs of samples randomly selected ensuring they belong to different classes
  - For each pair, via convex combination, 20 samples are synthesised — **total 20K samples**
  - The heat-map is then created
    - Intensity of (\lambda, H) bin indicates the number of samples in that bin

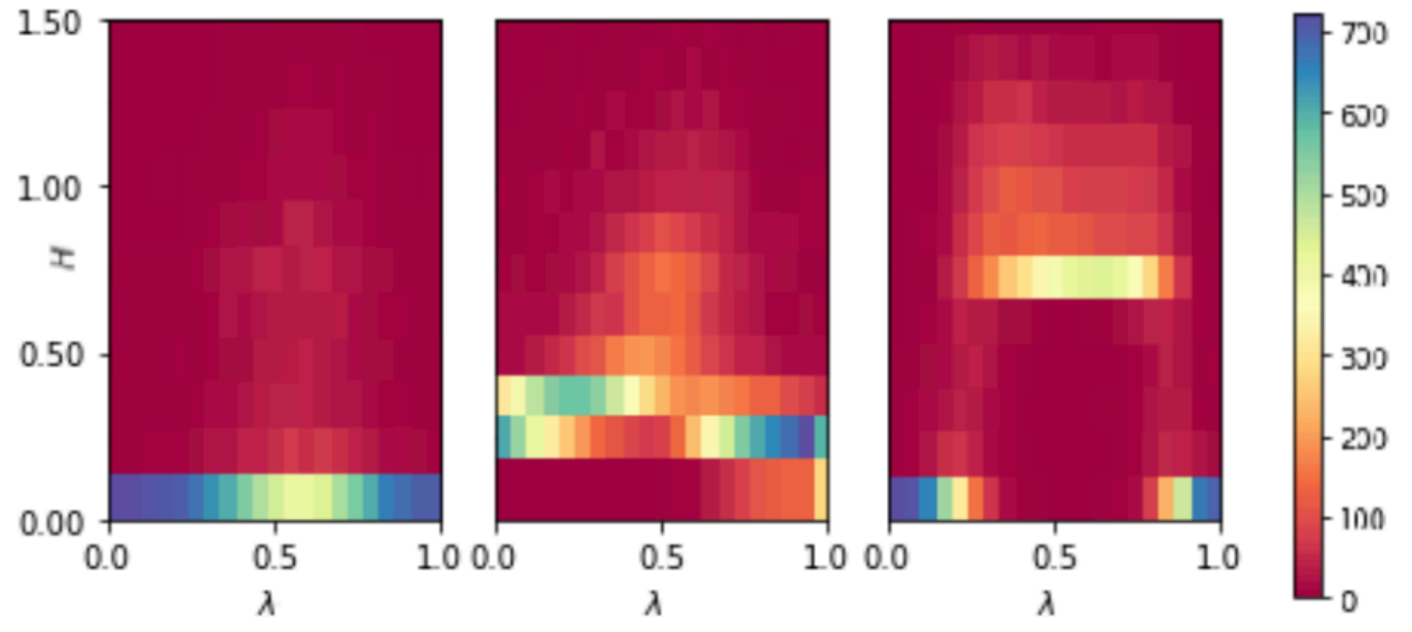


Figure 3: **Heatmaps of the entropy profiles** as the interpolation factor $\lambda \in [0, 1]$ between samples of two classes varies. Left (**DNN**), Middle (**Mixup**), Right (**RegMixup**). Note, RegMixup induces high-entropy barrier separating in-distribution & out-distribution samples.

(A relatively more regular behaviour of uncertainty w.r.t. input space for RegMixup)

Improving Calibration and Uncertainty Estimates

RegMixup (Experiments) — In-distribution and covariate shift

Methods	IND Accuracy: WRN28-10 C10 (Test) Accuracy (↑)	WRN28-10 C100 (Test) Accuracy (↑)	RN50 C10 (Test) Accuracy (↑)	RN50 C100 (Test) Accuracy (↑)
DNN	96.14	81.58	95.19	79.19
Mixup	97.01	82.60	96.05	80.12
RegMixup (**our**)	**97.46**	**83.25**	**96.71**	**81.52**
DNN-SN	96.22	81.60	95.20	79.27
DNN-SRN	96.22	81.38	95.39	78.96
SNGP	95.98	79.20	-	-
DUQ	94.7	-	-	-
KFAC-LLLA	96.11	81.56	95.21	79.41
Augmix	96.40	81.10	-	-
DE (5×)	96.75	83.85	96.23	82.09

Table 2: Accuracies (%) on IND samples for models trained on C10 and C100

Methods	Covariate Shift Accuracy: WRN28-10 C10-C Accuracy (↑)	WRN28-10 C10.1	WRN28-10 C10.2	WRN28-10 C100-C	R50 C10-C Accuracy (↑)	R50 C10.1	R50 C10.2	R50 C100-C
DNN	76.60	90.73	84.79	52.54	75.18	88.58	83.31	50.62
Mixup	81.68	91.29	86.55	56.99	78.63	90.03	84.61	53.96
RegMixup (**our**)	83.13	**92.79**	**88.05**	59.44	**81.18**	**91.58**	**86.72**	**57.64**
DNN-SN	76.56	91.01	84.72	52.61	74.88	88.26	82.96	50.55
DNN-SRN	77.21	90.88	85.24	52.54	75.40	88.61	83.49	50.48
SNGP	78.37	90.80	84.95	57.23	-	-	-	-
DUQ	71.6	-	-	50.4	-	-	-	-
KFAC-LLLA	76.56	90.68	84.68	52.57	75.18	88.34	83.50	50.85
AugMix	**90.02**	91.6	85.9	**68.15**	-	-	-	-
DE (× 5)	78.32	92.17	85.59	55.58	77.63	90.05	85.00	53.91

Table 3: Accuracies (%) on covariate shifted samples for models trained on C10 and C100.

Improving Calibration and Uncertainty Estimates

RegMixup (Experiments) — Out-of-distribution (AUROC)

	WRN28-10						RN50					
	CIFAR10 (In-Distribution)			CIFAR100 (In-Distribution)			CIFAR10 (In-Distribution)			CIFAR100 (In-Distribution)		
Out-of-Distribution	**C100**	**SVHN**	**T-ImageNet**	**C10**	**SVHN**	**T-ImageNet**	**C100**	**SVHN**	**T-ImageNet**	**C10**	**SVHN**	**T-ImageNet**
Methods	**AUROC (↑)**			**AUROC (↑)**			**AUROC (↑)**			**AUROC (↑)**		
DNN	88.61	96.00	86.44	81.06	79.68	80.99	88.61	93.20	87.82	79.33	82.45	79.89
Mixup	83.17	87.53	84.02	78.37	78.68	80.61	84.24	89.40	84.89	77.02	76.86	80.14
RegMixup (**our**)	89.63	**96.72**	**90.19**	**81.27**	**89.32**	**83.13**	**89.63**	**95.39**	**90.04**	**79.44**	**88.66**	**82.56**
DNN-SN	88.56	95.59	87.71	81.10	83.43	82.26	88.19	93.46	87.55	79.20	80.78	79.90
DNN-SRN	88.46	96.12	87.43	81.26	85.51	82.41	88.82	93.54	87.82	78.77	82.39	79.70
SNGP	**90.61**	95.25	90.01	79.05	86.78	82.60	-	-	-	-	-	-
KFAC-LLLA	89.33	94.17	87.81	81.04	80.32	81.57	89.54	93.13	88.32	79.30	82.80	80.17
Aug-Mix	89.78	91.3	88.99	81.10	76.64	80.56	-	-	-	-	-	-
DE (5×)	91.25	97.53	89.52	83.26	85.07	83.40	91.38	96.90	90.5	81.93	85.08	82.15

Table 5: Out-of-distribution detection results (%) for WideResNet28-10 and ResNet50 for models trained on C10 and C100. See Appendix B for the cross-validated hyperparameters.

Improving Calibration and Uncertainty Estimates

RegMixup (Experiments) — Calibration

Methods	WRN28-10 C10 (Test) AdaECE (↓)	WRN28-10 C100 (Test) AdaECE (↓)	RN50 C10 (Test) AdaECE (↓)	RN50 C100 (Test) AdaECE (↓)
DNN	1.34	3.84	1.45	2.94
Mixup	1.16	1.98	2.17	7.47
RegMixup (**our**)	**0.50**	**1.76**	**0.94**	**1.53**
SNGP	0.87	1.94	-	-
Augmix	1.67	5.54	-	-
DE (5×)	1.04	3.29	1.28	2.98

Table 6: CIFAR IND calibration performance (%).

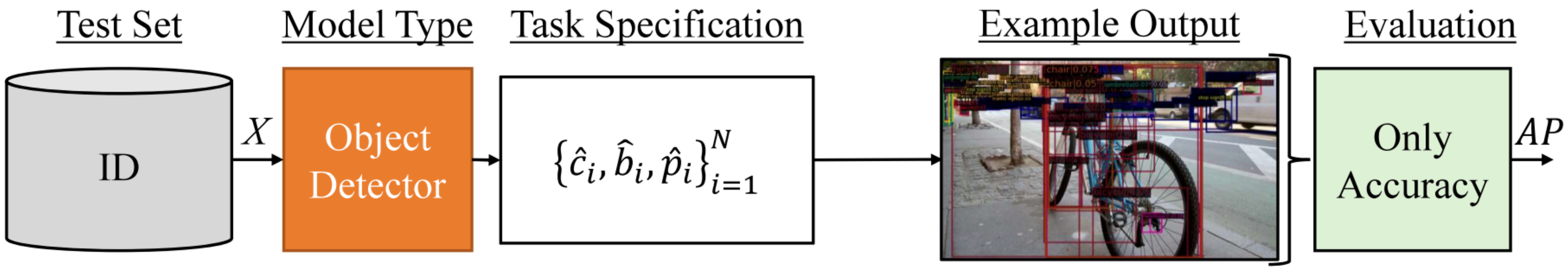
Methods	WRN28-10 C10-C	WRN28-10 C10.1	WRN28-10 C10.2	WRN28-10 C100-C	R50 C10-C	R50 C10.1	R50 C10.2	R50 C100-C
DNN	12.62	4.13	8.81	9.94	12.29	4.36	8.89	19.76
Mixup	7.93	4.39	7.44	10.45	**10.75**	5.72	10.59	12.63
RegMixup (**our**)	9.08	**2.57**	**6.83**	**7.93**	11.37	**2.89**	**6.74**	**11.47**
SNGP	11.34	4.36	8.33	10.43	-	-	-	-
AugMix	**4.56**	3.23	8.33	12.15	-	-	-	-
DE (5 ×)	10.31	2.60	7.50	12.36	12.68	4.10	6.94	12.36

Table 8: CIFAR CS calibration performance (%). (Covariate Shift; metric: AdaECE (↓))

Accuracy, Uncertainty estimates, and Calibration — all three aspects are improved

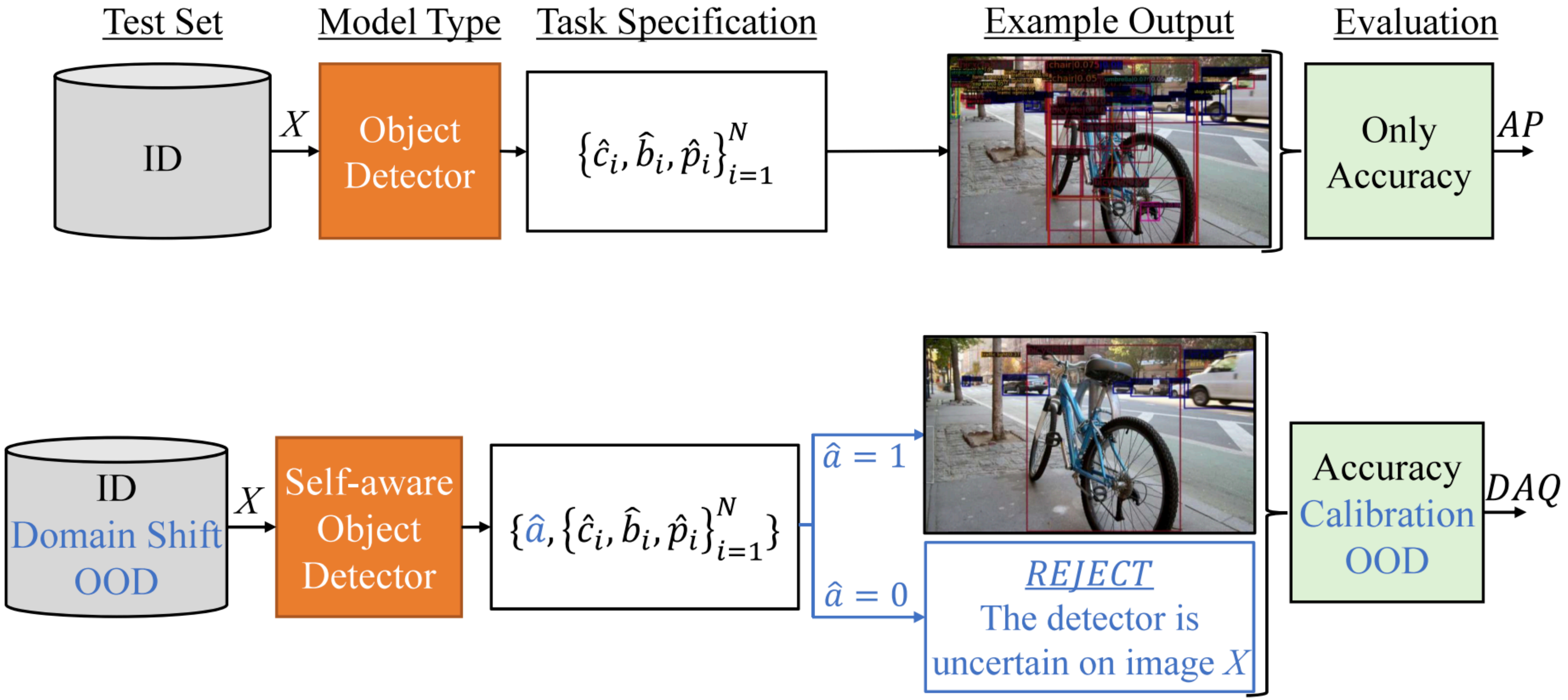
Calibrating Object Detectors

Self-Aware Object Detectors [Oksuz et al., CVPR23]



Calibrating Object Detectors

Self-Aware Object Detectors [Oksuz et al., CVPR23]



Calibrating Object Detectors

Self-Aware Object Detectors [Oksuz et al., CVPR23]

- Object detection is a **joint** task — classification and regression

$$\mathrm{LaECE}^c = \sum_{j=1}^{J} \frac{|\hat{\mathcal{D}}_j^c|}{|\hat{\mathcal{D}}^c|} \left| \bar{p}_j^c - \mathrm{precision}^c(j) \times \mathrm{I\bar{o}U}^c(j) \right|$$

Calibrating Object Detectors

Self-Aware Object Detectors [Oksuz et al., CVPR23]

- Object detection is a **joint** task — classification and regression

$$\mathrm{LaECE}^c = \sum_{j=1}^{J} \frac{|\hat{\mathcal{D}}_j^c|}{|\hat{\mathcal{D}}^c|} \left| \bar{p}_j^c - \mathrm{precision}^c(j) \times \bar{\mathrm{IoU}}^c(j) \right|$$

- Need a measure of uncertainty
  - Turns out object detectors are **very good at detecting OOD images** if the uncertainties are quantified using only top-3 predictions (1-confidence)
- Create dataset — using COCO, nuImages, Obj365, iNaturalist, etc.

Calibrating Object Detectors

Self-Aware Object Detectors [Oksuz et al., CVPR23]

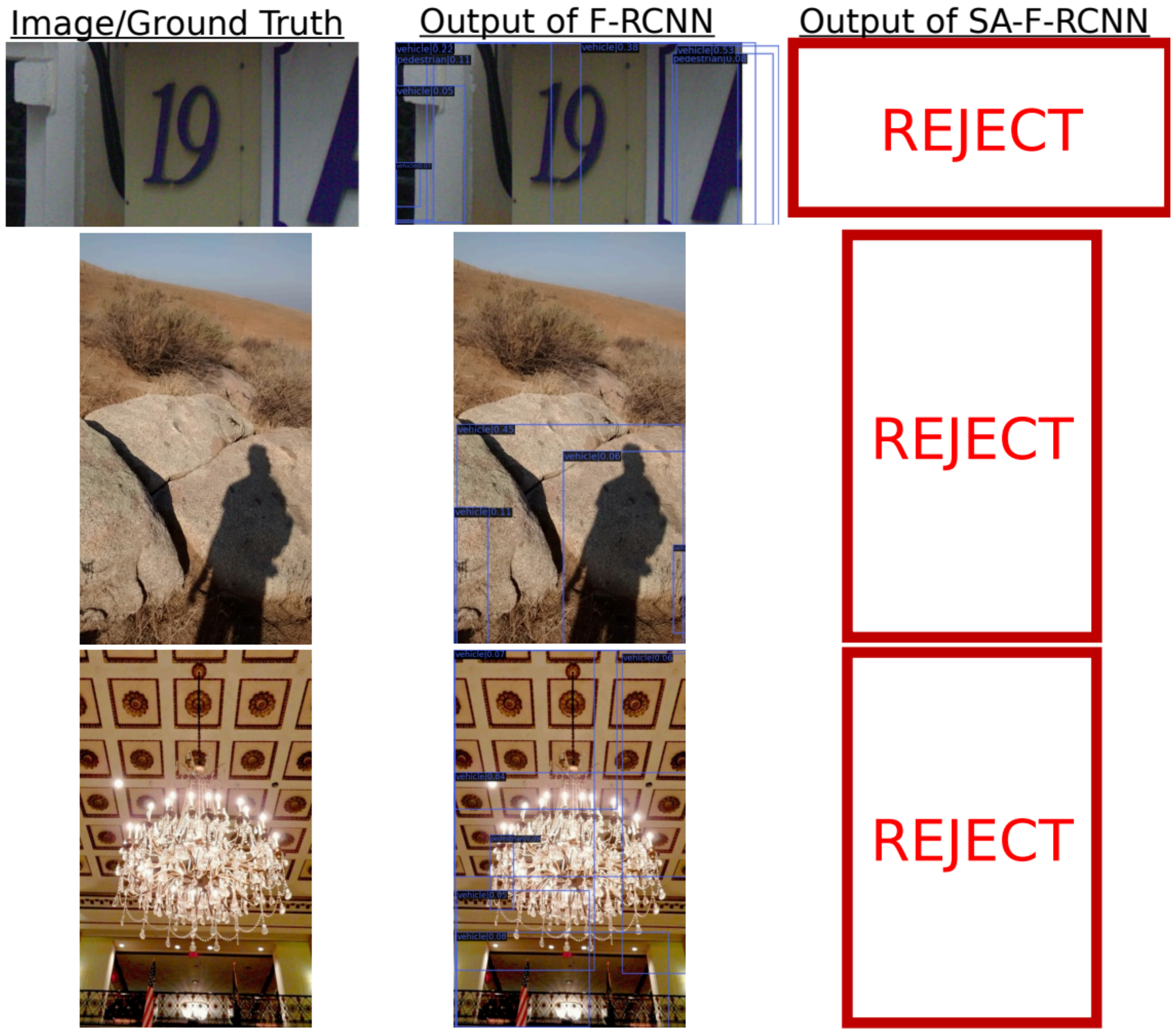


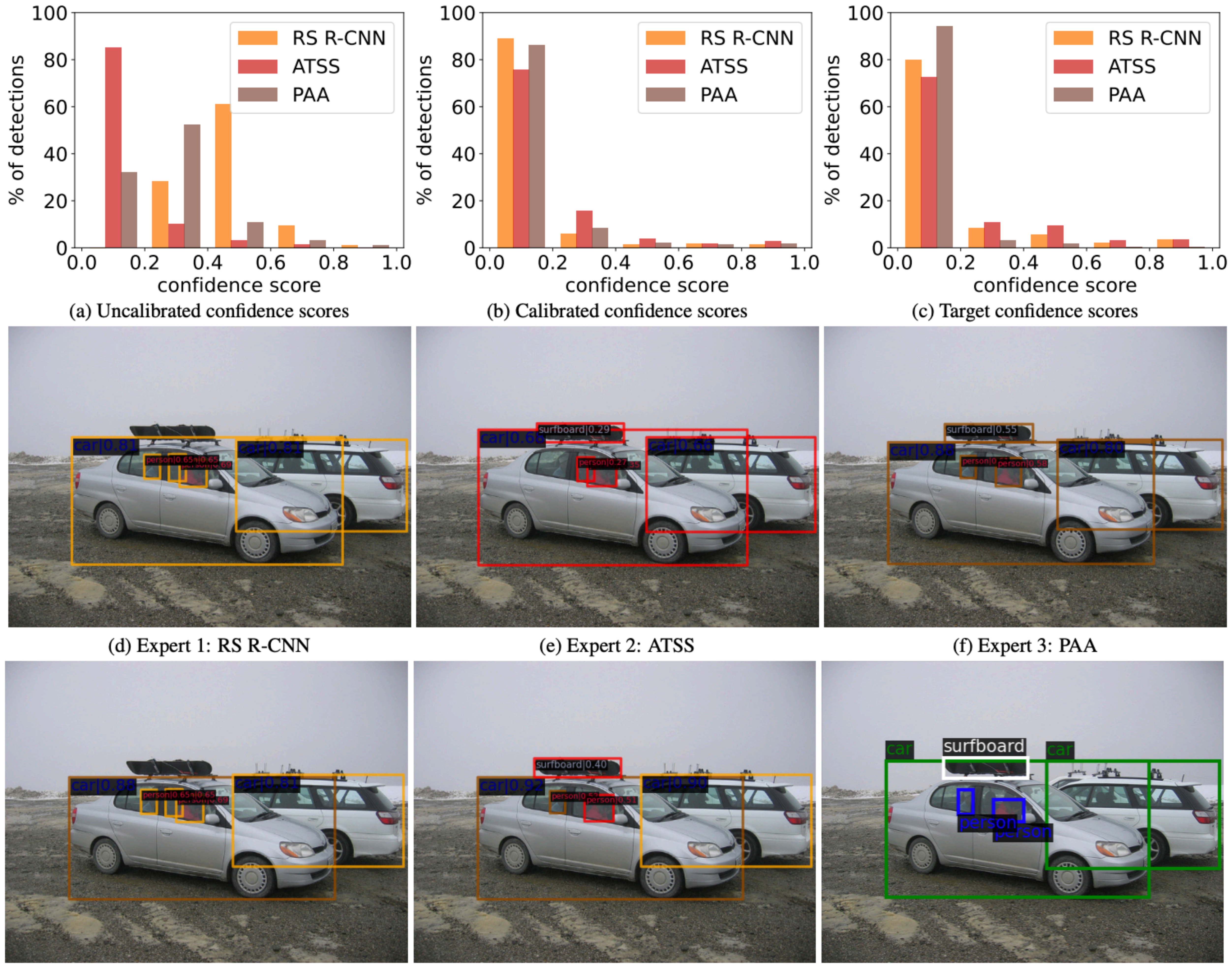
Figure A.17. Qualitative Results of F-RCNN vs. SA-F-RCNN on $\mathcal{D}_{\mathrm{OOD}}$. The images in the first, second and third rows correspond SVHN, iNaturalist and Objects365 subset of $\mathcal{D}_{\mathrm{OOD}}$. While F-RCNN performs inference with non-empty detections sets, SA-F-RCNN rejects all of these images properly.

Mixture of Calibrated Experts (MoCaE)

[Oksuz et al., arXiv 23]

- Calibration — IoU of True-positive boxes with ground-truth should match confidence

- Given a trained model, take a val-set, use Isotonic regression (or similar) to calibrate — that's it.



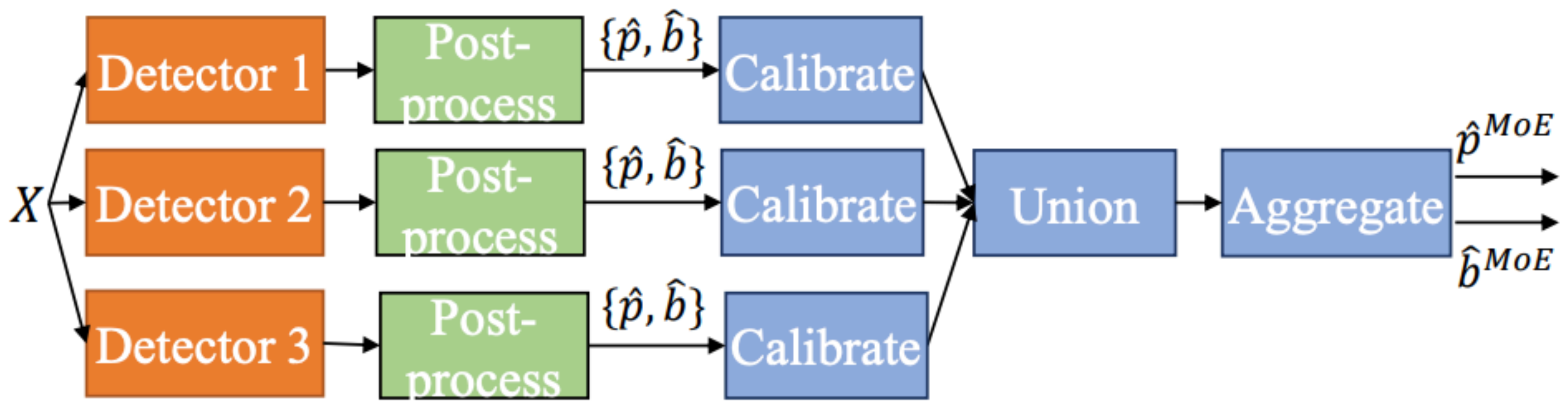
(a) Uncalibrated confidence scores (b) Calibrated confidence scores (c) Target confidence scores

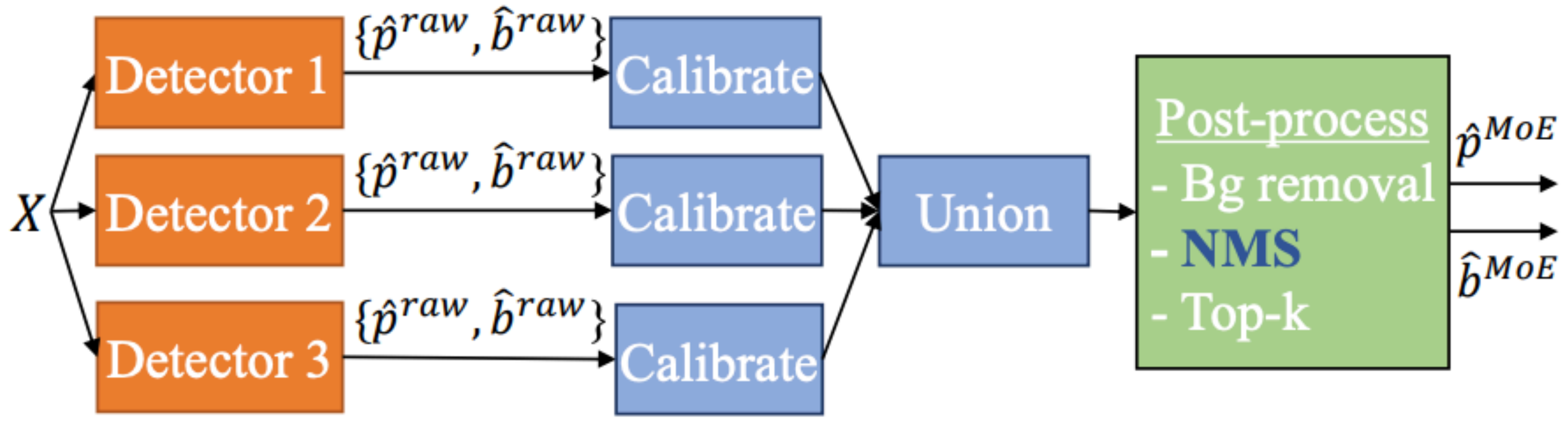
(d) Expert 1: RS R-CNN (e) Expert 2: ATSS (f) Expert 3: PAA

(g) Mixture of Uncalibrated Experts (h) Mixture of Calibrated Experts (i) Ground Truth Objects

Mixture of Calibrated Experts (MoCaE)

[Oksuz et al., arXiv 23]





Mixture of Calibrated Experts (MoCaE)

[Oksuz et al., arXiv 23]

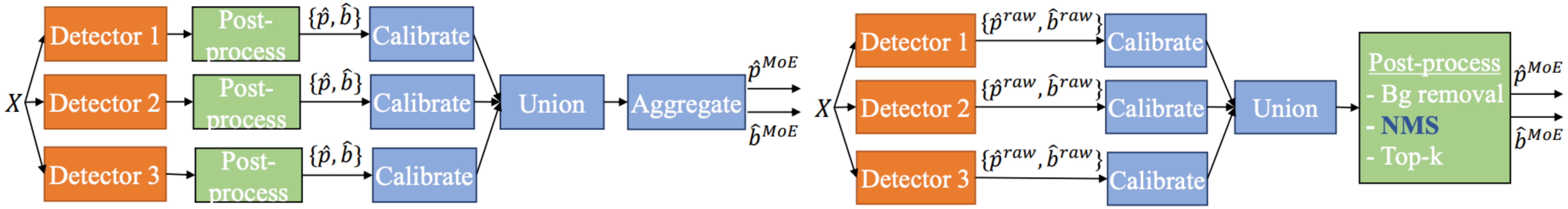


Table 4. Object detection performance on COCO *test-dev* and *mini-test* using strong object detectors. The gains are reported compared to the best single model as underlined. MoCaE maintains the significant AP boost also for this challenging setting as well.

Method	Pretraining Data	Backbone	COCO test-dev AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	COCO minitest AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv7 [60]	None	L-size conv.	55.5	73.0	60.6	37.9	58.8	67.7	55.6	73.1	60.6	41.2	60.4	69.5
QueryInst [17]	None	Swin-L	55.7	75.7	61.4	36.2	58.4	70.9	55.9	75.4	61.3	38.5	60.8	73.2
DyHead [13]	ImageNet22K	Swin-L	56.6	75.5	61.8	39.4	59.8	68.7	56.8	75.6	62.2	42.8	60.6	71.0
Vanilla MoE	N/A	N/A	57.6 +1.0	76.6 +0.9	63.2 +1.4	40.0 +0.6	60.9 +1.1	70.8 −0.1	57.7 +0.9	76.3 +0.7	62.9 +0.7	42.6 −0.2	62.7 +1.9	72.8 −0.4
MoCaE (Ours)	N/A	N/A	**59.0** +2.4	**77.2** +1.5	**64.7** +2.9	**41.1** +1.7	**62.6** +2.8	**72.4** +1.5	**58.9** +2.1	**76.8** +1.1	**64.3** +2.1	**44.7** +1.9	**63.6** +2.8	**74.1** +1.1

To Summarise

- Calibration is important
- But thinking about all the failure modes together is crucial (calibration, OOD, covariate-shift) as one might impact another — e.g., RegMixup
- Calibration for object detectors is quite open research problem with very limited work produced so far
- Do we need new architectures? e.g., Transformers?
  - Turns out Transformers suffer from similar vulnerabilities as CNNs [An Impartial Take …;Pinto et al., ECCV22]
- LLMs — open problem but without proper penetration testing and reliability certificates, these models will most likely not be used in any safety-critical situations. There is a good research opportunity here
- Multi-modality (CLIP)? Can we continually fine-tune with reliability guarantees? Similar to [Fine-tuning can cripple your foundation model;…;Mukhoti et al., arXiv23]

Thank you for your time

Questions?