Confidence Calibration and one of its applications

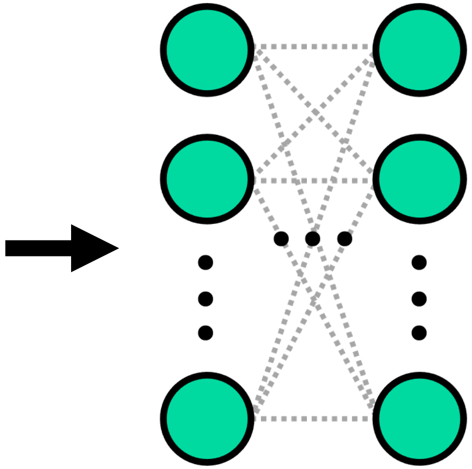
Sooyong Jang

STAT 991

Apr 19, 2022

Apr 21, 2022

ML and Confidence



Neural Network

Prediction:
"Persian cat"

Confidence:
99.78 %

Image: ImageNet

Good and bad confidence

Prediction: Tabby (51.60 %)
True label: Egyptian Cat

Prediction: Wallaby (97.96 %)
True label: Egyptian Cat

Prediction: Lynx (96.15 %)
True label: Tiger Cat

Images: ImageNet

Let’s make confidence accurate!

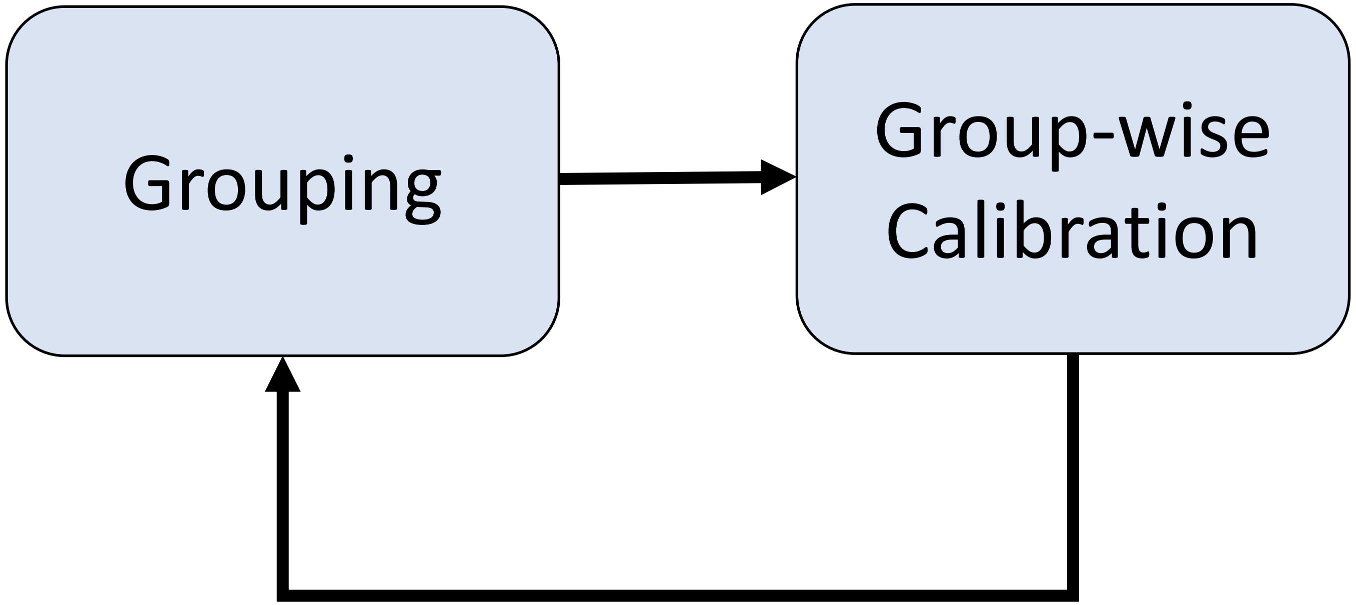
ReCal: Recursive Lossy Label-Invariant Calibration

Sooyong Jang, Insup Lee, James Weimer

Improving Classifier Confidence using Lossy Label-Invariant Transformations

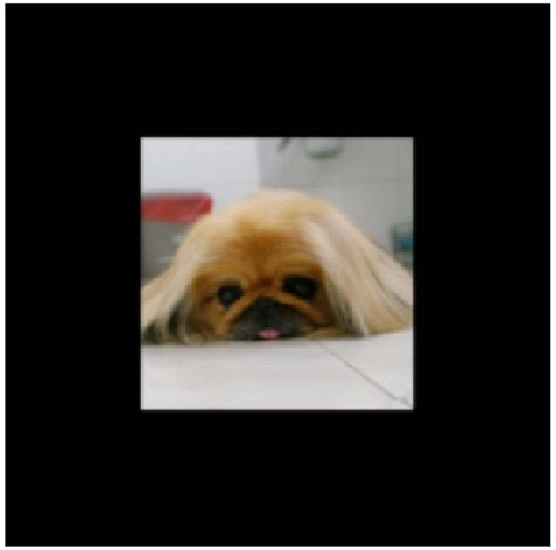
AISTATS 2021

Key Idea



Lossy Label-Invariant Transformation

0.5x
Zoom-out



Ground Truth: Pekinese
Prediction: Pekinese
Confidence: 99.10 %

Ground Truth: Pekinese
Prediction: Pekinese
Confidence: 97.99 %

Prediction **DOES NOT** change
Confidence **DOES** decrease

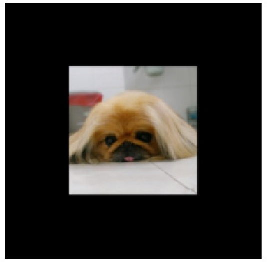
*Some transformations yield an expected trend in prediction **AND** confidence!!!*

Grouping Examples by Confidence and Prediction

Given an image and its transform w/ corresponding predictions and confidence …

$\xrightarrow{f} (\hat{y}, \hat{p})$

Transformation



$\xrightarrow{f} (\bar{y}, \bar{p})$

… use real-world intuition to form groups

Confidence

Predictions	Increases $\bar{p} \geq \hat{p}$	Decreases $\bar{p} < \hat{p}$
different $\hat{y} \neq \bar{y}$	**inconsistent** *needs **most** adjustment*	**mixed consistency** *needs **more** adjustment*
same $\hat{y} = \bar{y}$	**mixed consistency** *needs **more** adjustment*	**consistent** *needs **least** adjustment*

Group-wise Calibration

Temperature Scaling
Find c^1

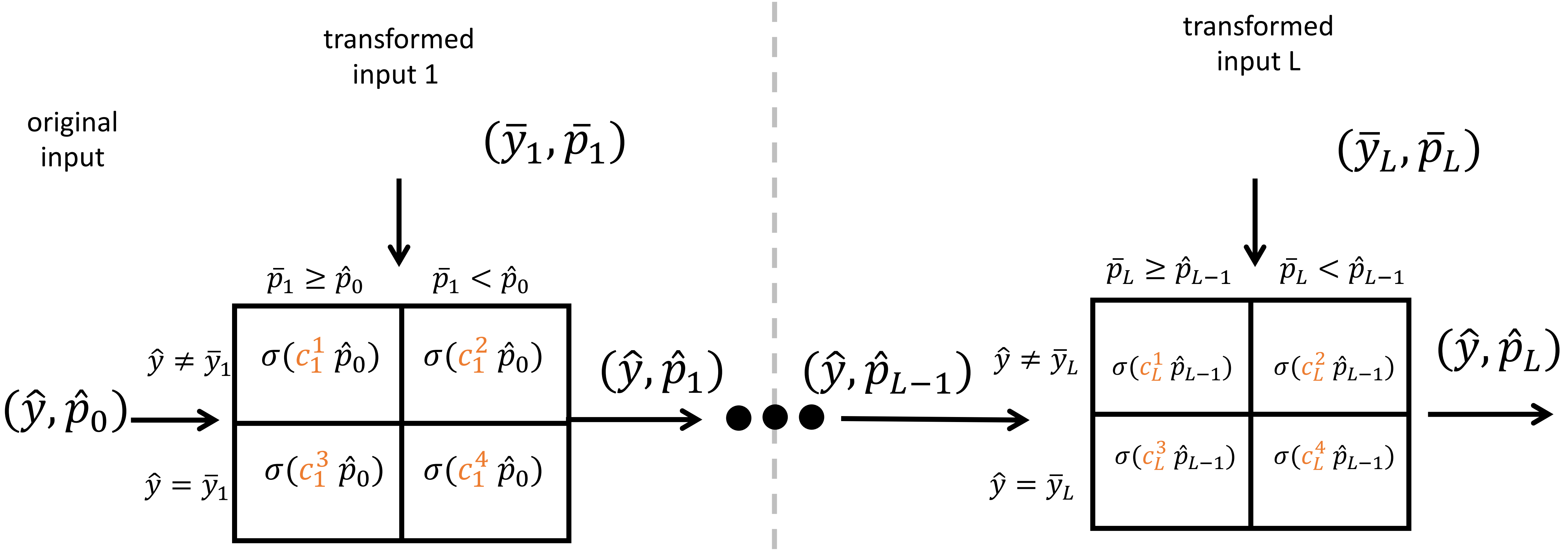
Temperature Scaling
Find c^2

	Confidence Increases $\bar{p} \geq \hat{p}$	**Confidence** Decreases $\bar{p} < \hat{p}$
Predictions different $\hat{y} \neq \bar{y}$	**inconsistent** *needs **most** adjustment*	**mixed consistency** *needs **more** adjustment*
Predictions same $\hat{y} = \bar{y}$	**mixed consistency** *needs **more** adjustment*	**consistent** *needs **least** adjustment*

Temperature Scaling
Find c^3

Temperature Scaling
Find c^4

ReCal using Lossy Label-Invariant Transformations



Temperature scaling calibration for each group trained on validation data (1 parameter: c)

Key observations / features:

- Predictions remain unchanged (i.e., accuracy preserved)
- 4*L parameters to learn (i.e., small confidence model size)

Which Lossy Label-Invariant Transformations?

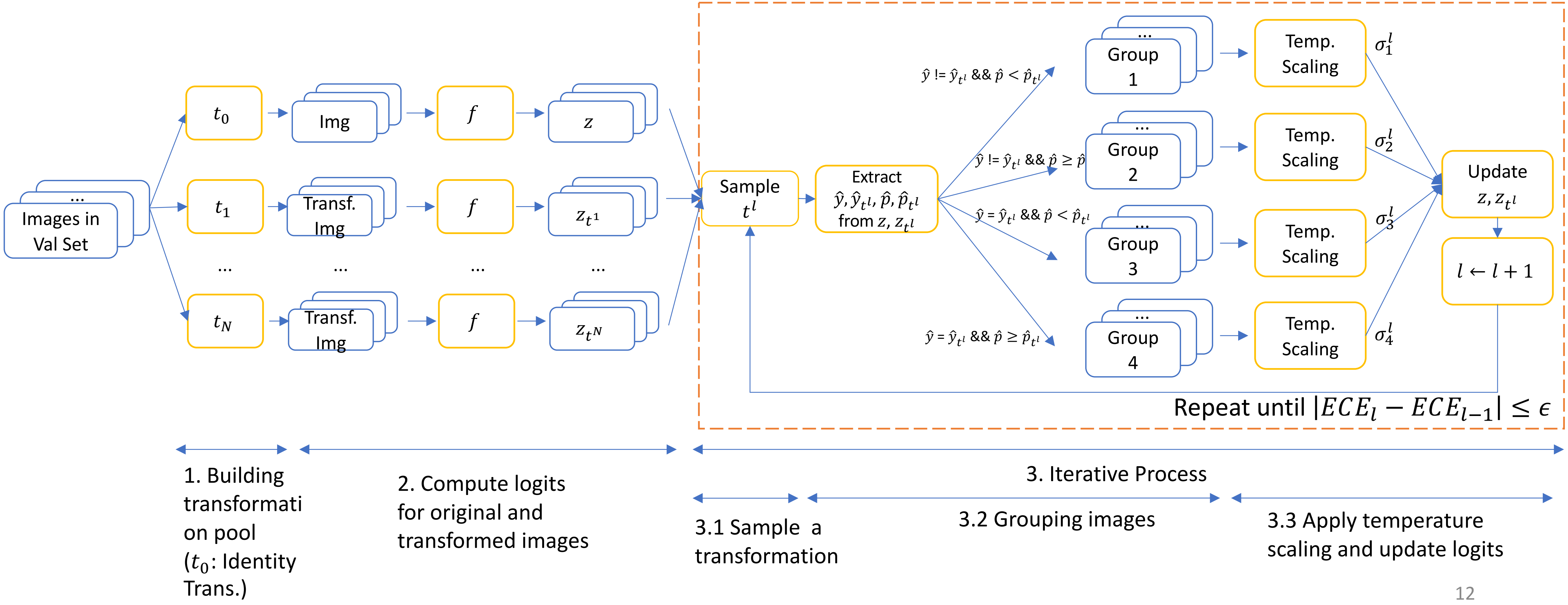
1. Specify transformations pool
   - Transformation Type
     - Zoom-out, Brightness, …
   - Parameter Range
     - (0.1, 0.9), (0.5, 0.9), …
   - Number of Transformation
     - 10, 20, …

   E.g., [(Zoom-out, 0.36), (Zoom-out, 0.7), (Zoom-out, 0.85), (Zoom-out, 0.23), (Zoom-out, 0.15)]
2. Randomly sample a transformation for each iterations

ReCal – Design Time

Benefits:

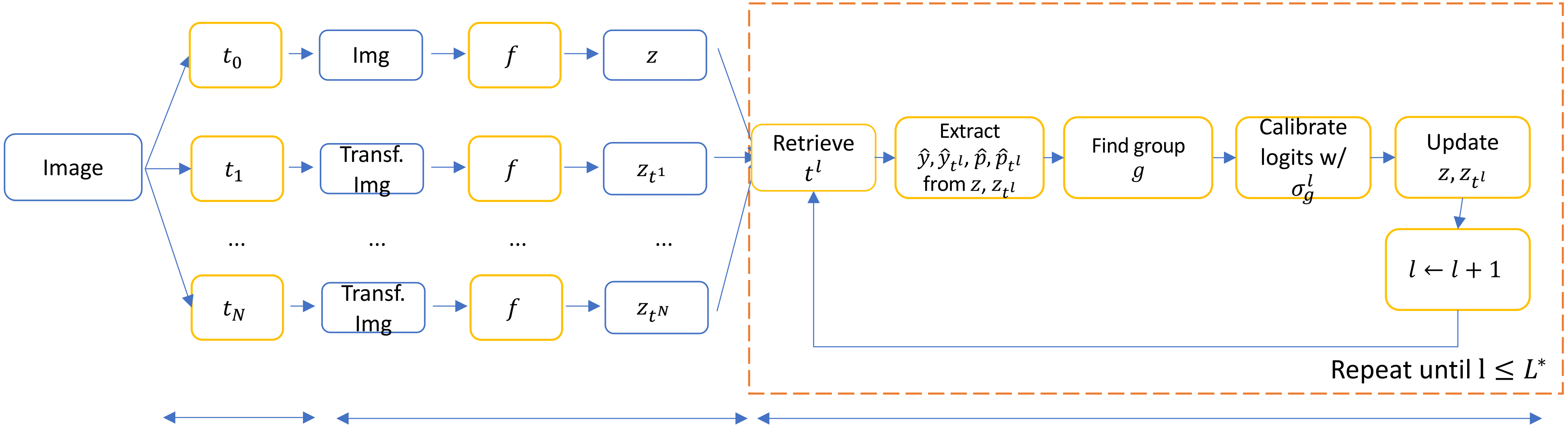
- Only need to perform each transformation once
- Only 4 parameters per loop



ReCal – Run Time

Benefits:

- Fast to compute – i.e., scales well



Building transformation pool (t_0: Identity Transformation)

Compute logits for original and transformed images

Iterative Process

Results: Brier Score

Dataset	Model	Uncal.	TS	VS	MS-ODIR	Dir-ODIR	ReCal (z, .1-.9, 20)	ReCal (z, .5-.9, 10)	ReCal (b, .1-.9, 20)
CIFAR10	DenseNet40	0.013585	0.012330	0.012300	0.012256	0.012296	**0.012225**	0.012231	0.012324
CIFAR10	LeNet5	0.037836	0.037792	0.037748	0.037745	0.037706	**0.037395**	0.037403	0.037784
CIFAR10	ResNet110	0.011537	0.010439	0.010378	0.010382	0.010350	0.010322	**0.010317**	0.010441
CIFAR10	ResNet110 SD	0.015472	0.014395	0.014325	0.014231	0.014302	0.014212	**0.014140**	0.014425
CIFAR10	WRN 28-10	0.006731	0.006357	0.006380	0.006342	0.006336	**0.006300**	0.006344	0.006363
CIFAR100	DenseNet40	0.004862	0.004329	0.004346	0.004333	0.004318	0.004304	**0.004302**	0.004332
CIFAR100	LeNet5	0.007581	0.007588	0.007587	0.007580	0.007567	0.007557	**0.007543**	0.007581
CIFAR100	ResNet110	0.004521	0.004144	0.004180	0.004178	0.004149	0.004130	**0.004119**	0.004149
CIFAR100	ResNet110 SD	0.004344	0.004064	0.004046	0.004045	0.004047	0.004035	**0.004028**	0.004067
CIFAR100	WRN 28-10	0.002929	0.002915	0.002948	0.002901	**0.002898**	0.002913	0.002913	0.002926
ImageNet	DenseNet161	0.000323	0.000319	0.000316	**0.000313**	0.000324	0.000318	0.000319	0.000319
ImageNet	ResNet152	0.000305	0.000302	0.000301	**0.000299**	0.000307	0.000302	0.000302	0.000302

Results: ECE

Dataset	Model	Uncal.	TS	VS	MS-ODIR	Dir-ODIR	ReCal (z, .1-.9, 20)	ReCal (z, .5-.9, 10)	ReCal (b, .1-.9, 20)
CIFAR10	DenseNet40	0.052026	0.007037	0.004438	0.005161	**0.003943**	0.010143	0.008721	0.005892
CIFAR10	LeNet5	0.018170	0.011963	**0.009174**	0.014147	0.010525	0.011785	0.010507	0.010669
CIFAR10	ResNet110	0.045646	0.008770	0.009442	0.008829	0.008366	0.008986	**0.008206**	0.009177
CIFAR10	ResNet110 SD	0.053770	0.011407	**0.008552**	0.010187	0.009369	0.011973	0.012103	0.012845
CIFAR10	WRN 28-10	0.025076	0.009709	0.009564	0.009175	0.009429	**0.009092**	0.012459	0.010261
CIFAR100	DenseNet40	0.172838	0.015435	0.026634	0.029628	0.018949	0.015398	**0.011713**	0.018059
CIFAR100	LeNet5	**0.009991**	0.021064	0.015524	0.013149	0.014172	0.019196	0.018426	0.019367
CIFAR100	ResNet110	0.142223	0.009101	0.029982	0.034519	0.023109	0.012142	**0.008487**	0.010614
CIFAR100	ResNet110 SD	0.122932	0.009310	0.035832	0.035478	0.020747	0.009987	0.014375	**0.007918**
CIFAR100	WRN 28-10	0.053396	0.043703	0.045178	0.035509	**0.034604**	0.037270	0.035279	0.035435
ImageNet	DenseNet161	0.056384	0.019873	0.023286	0.036785	0.047707	**0.013348**	0.014474	0.016981
ImageNet	ResNet152	0.049142	0.020069	0.020672	0.034736	0.039748	0.013869	**0.013491**	0.017483

ReCal performs very well – AND – Scales!!!

Results: Time for learning calibration function

(Unit: seconds)

Dataset	Model	TS	VS	MS-ODIR	Dir-ODIR	ReCal (z, .1-.9, 20)
CIFAR10	DenseNet40	**2.94**	<u>31.10</u>	77353.63	43001.99	84.04
CIFAR10	LeNet5	**1.86**	<u>12.06</u>	42830.58	37001.63	110.79
CIFAR10	ResNet110	**2.21**	<u>26.65</u>	70702.87	45836.87	38.85
CIFAR10	ResNet110 SD	**4.35**	<u>26.52</u>	85859.16	54783.42	58.74
CIFAR10	WRN 28-10	**7.68**	<u>28.22</u>	67955.20	36386.26	49.62
CIFAR100	DenseNet40	**14.03**	<u>26.31</u>	320284.77	134317.54	136.23
CIFAR100	LeNet5	**9.63**	<u>26.10</u>	109645.75	83324.48	97.77
CIFAR100	ResNet110	**8.63**	<u>26.61</u>	300360.19	134317.54	97.29
CIFAR100	ResNet110 SD	**13.24**	<u>26.73</u>	276767.31	126100.97	604.12
CIFAR100	WRN 28-10	**14.23**	<u>25.60</u>	161327.35	85532.50	125.84
ImageNet	DenseNet161	<u>865.40</u>	**285.73**	379487.45	276553.98	50730.17
ImageNet	ResNet152	<u>754.51</u>	**342.50**	215746.16	229493.41	71254.34

Transformation Selection

Data type decides transformation type.

Zoom-out,
Brightness,
Blur, Noises, …

Random Data Drop

Data size decides transformation parameter range.

More zoom-out
E.g., 0.1-0.9

Less zoom-out
E.g., 0.5-0.9

Summary

- ReCal uses Lossy Label-Invariant transformations to group inputs.
- ReCal improves the confidence calibration.

Why do we need calibrated confidence?

Apr 21, 2022

Calibration

Low ECE …

?

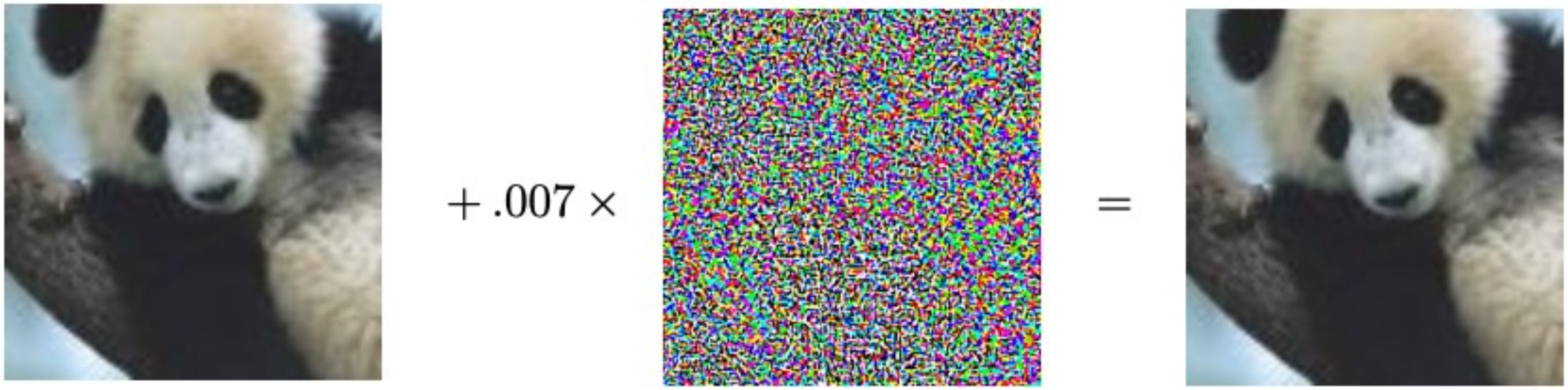
Confidence Calibrated Adversarial Training

David Stutz, Matthias Hein, Bernt Schiele

Confidence-Calibrated Adversarial Training: Generalizing to Unseen Attacks

ICML 2020

Adversarial Examples



Panda

Gibbon

Image: Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

Adversarial Training (AT)

Find $x + \delta$ which maximize loss $\mathcal{L}$ with respect to true label y

$$\min_{w} \mathbb{E}\left[\max_{||\delta||_\infty \leq \epsilon} \mathcal{L}(f(x+\delta; w), y)\right]$$

Find weights ω which minimize loss $\mathcal{L}$

Key Idea

Low Confidence on adv. examples

⬇

Uniform distribution: $\frac{1}{K}$

Target distribution for adv. examples

$$\tilde{y} = \lambda(\delta) \text{ one\_hot}(y) + \left(1 - \lambda(\delta)\right) \frac{1}{K}$$

Convex combination of one-hot distribution and uniform

$$\lambda(\delta) := \left(1 - \min\left(1, \frac{\|\delta\|_\infty}{\epsilon}\right)\right)^\rho$$

Another difference - Attack

With respect to ANY OTHER LABEL

$$\max_{\|\delta\|_\infty \leq \epsilon} \boxed{\max_{k \neq y} f_k(x+\delta; w)}$$

Attack for AT: $$\max_{\|\delta\|_\infty \leq \epsilon} \boxed{\mathcal{L}(f(x+\delta; w), y)}$$

With respect to TRUE LABEL

Algorithm

while true **do**
  choose random batch $(x_1, y_1), \ldots, (x_B, y_B)$.
  for $b = 1, \ldots, {}^{B}/_{2}$ **do**
    $\delta_b := \underset{\|\delta\|_\infty \leq \epsilon}{\operatorname{argmax}} \; \max_{k \neq y_b} f_k(x_b + \delta)$ (Eq. (4))
    $\tilde{x}_b := x_b + \delta_b$
    $\lambda(\delta_b) := (1 - \min(1, {}^{\|\delta_b\|_\infty}/_{\epsilon}))^\rho$ (Eq. (6))
    $\tilde{y}_b := \lambda(\delta_b)\, \text{one\_hot}(y_b) + (1 - \lambda(\delta_b)) \frac{1}{K}$ (Eq. (5))
  end for
  update parameters using Eq. (3):
    $\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b) + \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)$
end while

Adv. examples

Clean examples

How can we use calibrated confidence?

↓

Low confidence on adv. examples

↓

Let us reject low confidence examples

How low is enough? (0.5? 0.3?)

Find Threshold using hold-out set based on TPR

Results - MNIST

MNIST:	Err ↓ in %		*confidence-thresholed* **RErr** ↓ for τ@99%TPR					
	(clean) $\tau = 0$	(clean) 99%TPR	L_∞ $\epsilon = 0.3$	L_∞ $\epsilon = 0.4$	L_2 $\epsilon = 3$	L_1 $\epsilon = 18$	L_0 $\epsilon = 15$	adv. frames
	(seen)	(seen)	seen	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**
Normal	0.4	0.1	100.0	100.0	100.0	100.0	92.3	87.7
AT-50%	0.5	**0.0**	**1.7**	100.0	81.5	24.6	23.9	73.7
AT-100%	0.5	**0.0**	**1.7**	100.0	84.8	21.3	**13.9**	62.3
CCAT	**0.3**	0.1	7.4	**11.9**	**0.3**	**1.8**	14.8	**0.2**
* MSD	1.8	0.9	34.3	98.9	59.2	55.9	66.4	8.8
* TRADES	0.5	0.1	4.0	99.9	44.3	9.0	35.5	**0.2**

Results - SVHN

SVHN:	**Err** $\downarrow$ in %		*confidence-thresholed* **RErr** $\downarrow$ for τ@99%TPR					
	(clean) $\tau = 0$	(clean) 99%TPR	L_∞ $\epsilon = 0.03$	L_∞ $\epsilon = 0.06$	L_2 $\epsilon = 2$	L_1 $\epsilon = 24$	L_0 $\epsilon = 10$	adv. frames
	(seen)	(seen)	seen	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**
Normal	3.6	2.6	99.9	100.0	100.0	100.0	83.7	78.7
AT-50%	3.4	2.5	56.0	88.4	99.4	99.5	73.6	33.6
AT-100%	5.9	4.6	48.3	87.1	99.5	99.8	89.4	26.0
CCAT	**2.9**	**2.1**	**39.1**	**53.1**	**29.0**	**31.7**	**3.5**	**3.7**
* LID	3.3	2.2	91.0	93.1	92.2	90.0	41.6	89.8
* MAHA	3.3	2.2	73.0	79.5	78.1	67.5	41.5	9.9

Results – CIFAR10

CIFAR10:	**Err** ↓ in %		*confidence-thresholed* **RErr** ↓ for τ@99%TPR					
	(clean) $\tau = 0$	(clean) 99%TPR	L_∞ $\epsilon = 0.03$	L_∞ $\epsilon = 0.06$	L_2 $\epsilon = 2$	L_1 $\epsilon = 24$	L_0 $\epsilon = 10$	adv. frames
	(seen)	(seen)	seen	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**
Normal	8.3	7.4	100.0	100.0	100.0	100.0	84.7	96.7
AT-50%	16.6	15.5	62.7	93.7	98.4	98.4	74.4	78.7
AT-100%	19.4	18.3	59.9	90.3	98.3	98.0	72.3	79.6
CCAT	10.1	6.7	68.4	92.4	**52.2**	**58.8**	**23.0**	**66.1**
* MSD	18.4	17.6	53.2	89.4	88.5	68.6	39.2	82.6
* TRADES	15.2	13.2	**43.5**	**81.0**	70.9	96.9	36.9	72.1
* AT-Madry	13.0	11.7	45.1	84.5	98.7	97.8	42.3	73.3
* LID	**6.4**	**4.9**	99.0	99.2	70.6	89.4	47.0	**66.1**
* MAHA	**6.4**	**4.9**	94.1	95.3	90.6	97.6	49.8	70.0

Conclusions

- CCAT returns low confidence on adversarial examples.
- Using accurate confidence, adversarial examples can be rejected.

Thank you.
Any Questions?