Uncertainty and its Applications in Deep Neural Networks

Yonatan Geifman

Motivation

- Would you let a 95% accuracy CNN examine your MRI scans?

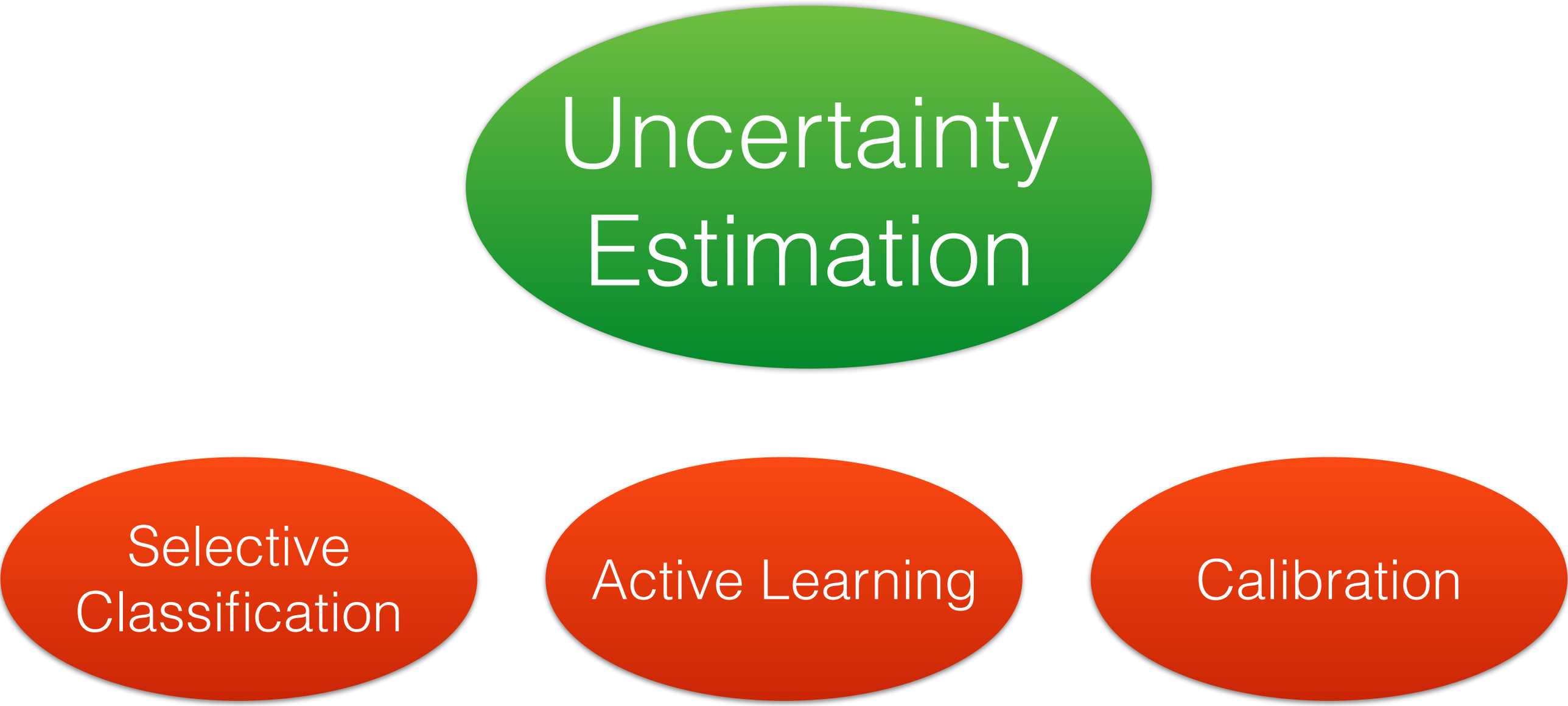
- Would you let a 54% accuracy classifier invest your funds?

- Would you fall a sleep in a 99% accurate autonomous car?

Outline

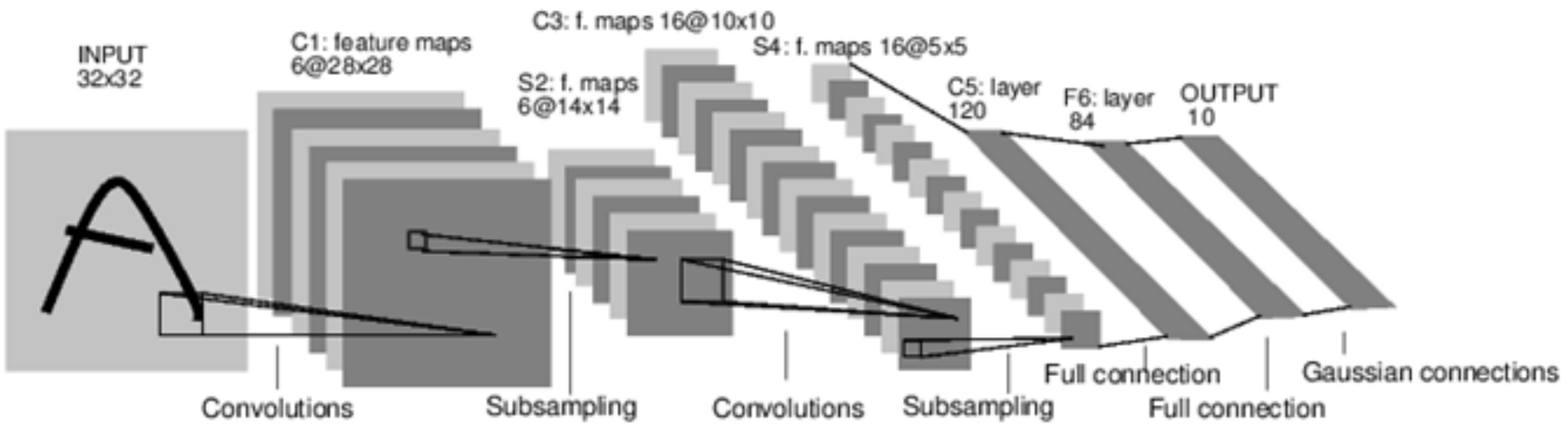
- Preliminaries and definitions
- Bias-reduced uncertainty estimation
- Selective classification

Uncertainty



Deep Neural Networks

- Multiple layers of processing units
- Feature representations learned at each layer
- Low level features to high level
- In this work we focus on convolutional neural networks for classification



A Full Convolutional Neural Network (LeNet)

Confidence Rate Functions

- For a classifier f ,We seek for a confidence rate function κ_f that reflects loss monotonicity

$$\kappa(x_1, \hat{y}_f(x)|f) \leq \kappa(x_2, \hat{y}_f(x)|f) \iff Pr_P[\hat{y}_f(x_1) \neq y_1] \geq Pr_P[\hat{y}_f(x_2) \neq y_2]$$

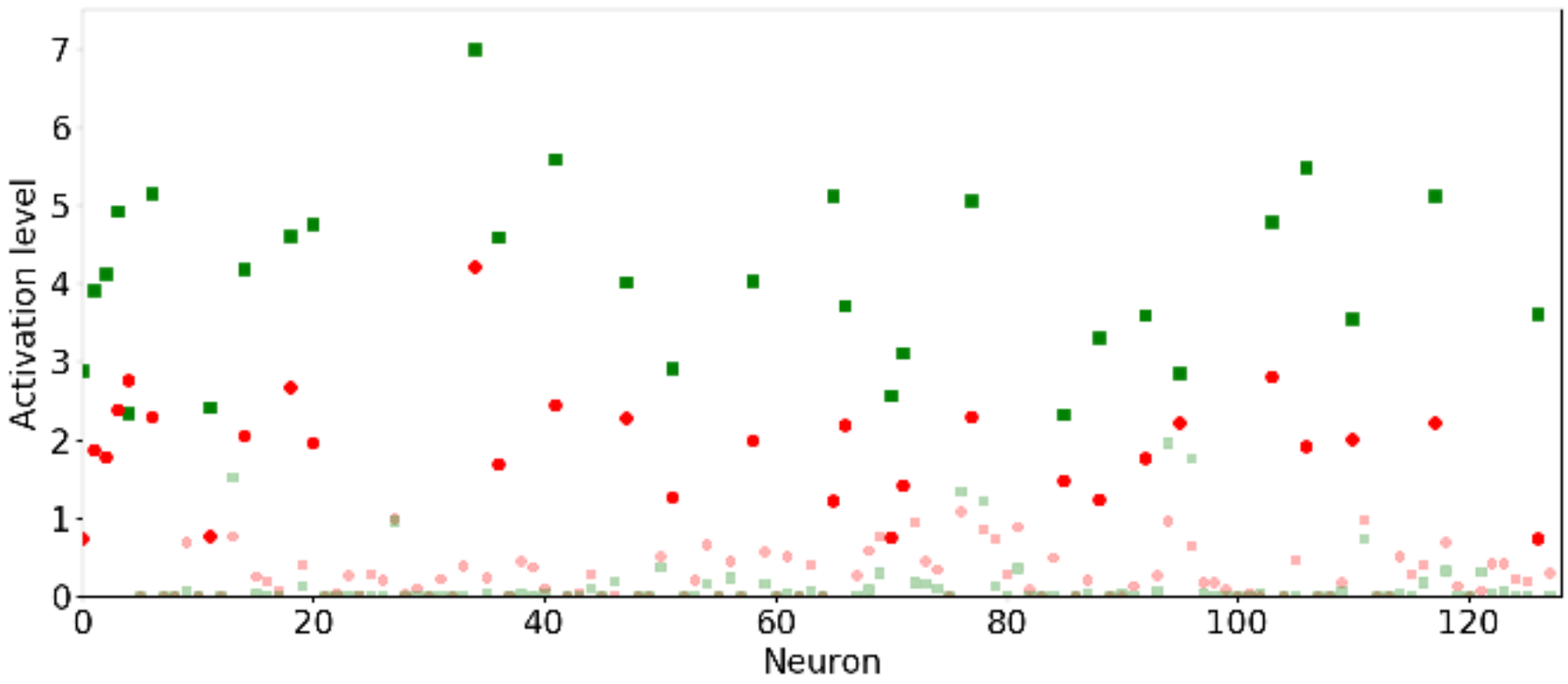
- We discuss three candidates:
  - Softmax response
  - MC-Dropout
  - Nearest neighbors distance

Confidence - Softmax Response

- Simply take κ to be the Softmax output

$$\kappa_f \triangleq \max_{j \in \mathcal{Y}} (f(x|j))$$

- Motivation - MNIST activations:



Confidence - MC-Dropout

- Apply dropout at inference
- Estimate prediction variance over numerous (100) forward passes with dropout (p=0.5)
- Intuition - kind of ensemble variance

Gal, Yarin, and Zoubin Ghahramani. "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning."

Confidence - NN distance

- Run nearest neighbors on the embedding space
- Extract scores from in-class vs out-class distances among the k nearest neighbors

Mandelbaum, Amit, and Daphna Weinshall. "Distance-based Confidence Score for Neural Network Classifiers." *arXiv preprint arXiv:1709.09844* (2017).

Confidence scores evaluation

- Confidence scores are:
  - Not probabilities
  - Might be unbounded (MC-dropout)
  - Defined across a set of instances
- Thus we need a way to evaluate a score based on a labeled set (empirical measure).
- We use selective prediction to asses confidence scores.

Selective Classification

- Selective Classifier is a pair (f, g)

$$(f,g)(x) = \begin{cases} f(x), & \text{if } g(x) = 1; \\ \text{don't know}, & \text{if } g(x) = 0. \end{cases}$$

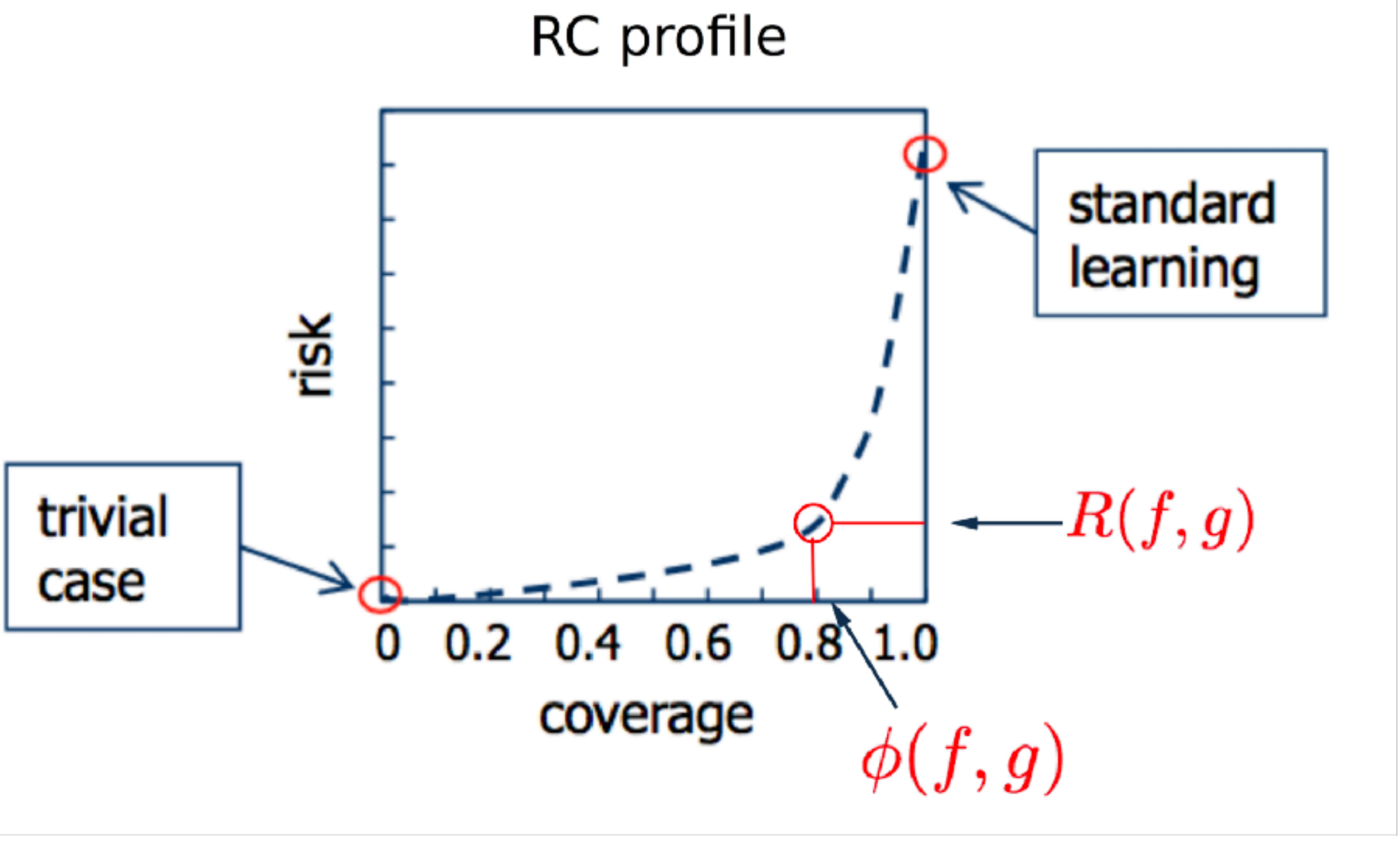
- Coverage:

$$\phi(f,g) \triangleq E_P[g(x)]$$

- Risk: $$R(f,g) \triangleq \frac{E_P[\ell(f(x),y)g(x)]}{\phi(f,g)}.$$

Ran El-Yaniv, and Yair Wiener. "On the foundations of noise-free selective classification."

Selective Classification



From kappa to selective classifier

- A selective classifier obtained by thresholding the confidence rate function

$$g_\theta(x) \triangleq \begin{cases} 1, & \text{if } \kappa_f(x) \geq \theta; \\ 0, & \text{otherwise.} \end{cases}$$

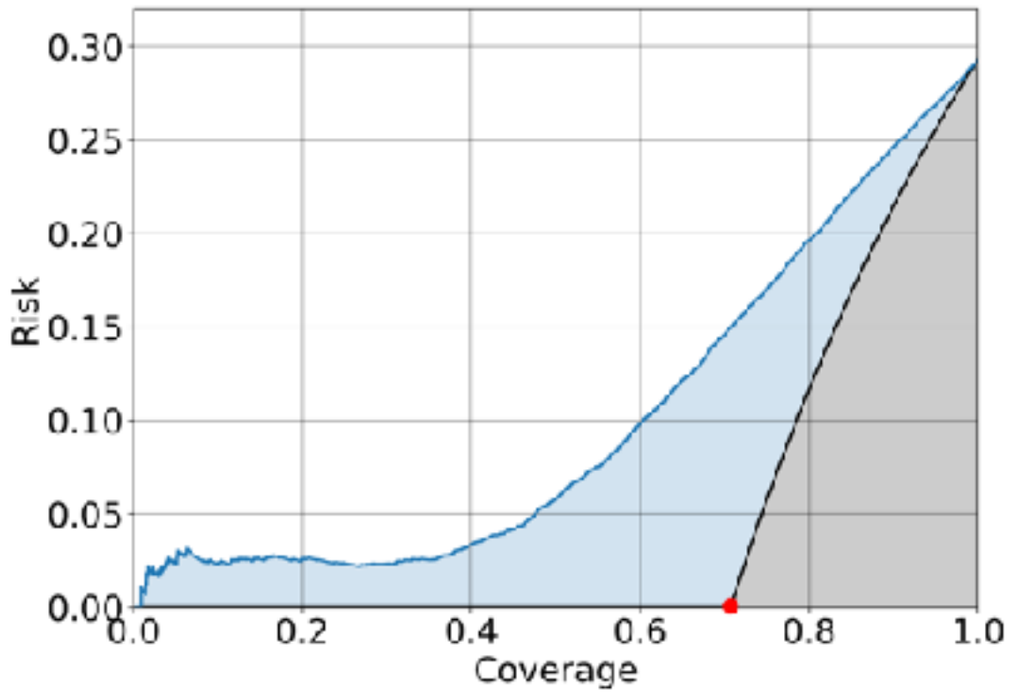
- Given a set S_m, we can derive a family of g functions based on all the possible thresholds θ.

Area under risk coverage curve

- Given a labeled set S_m , and a κ function, we define the AURC to be the area under the RC-curve.

- For ease of quantitive assessment we normalize it by the AURC of the best κ in hindsight. (E-AURC)

- Notation - $\text{E-AURC}(\kappa, f|S)$



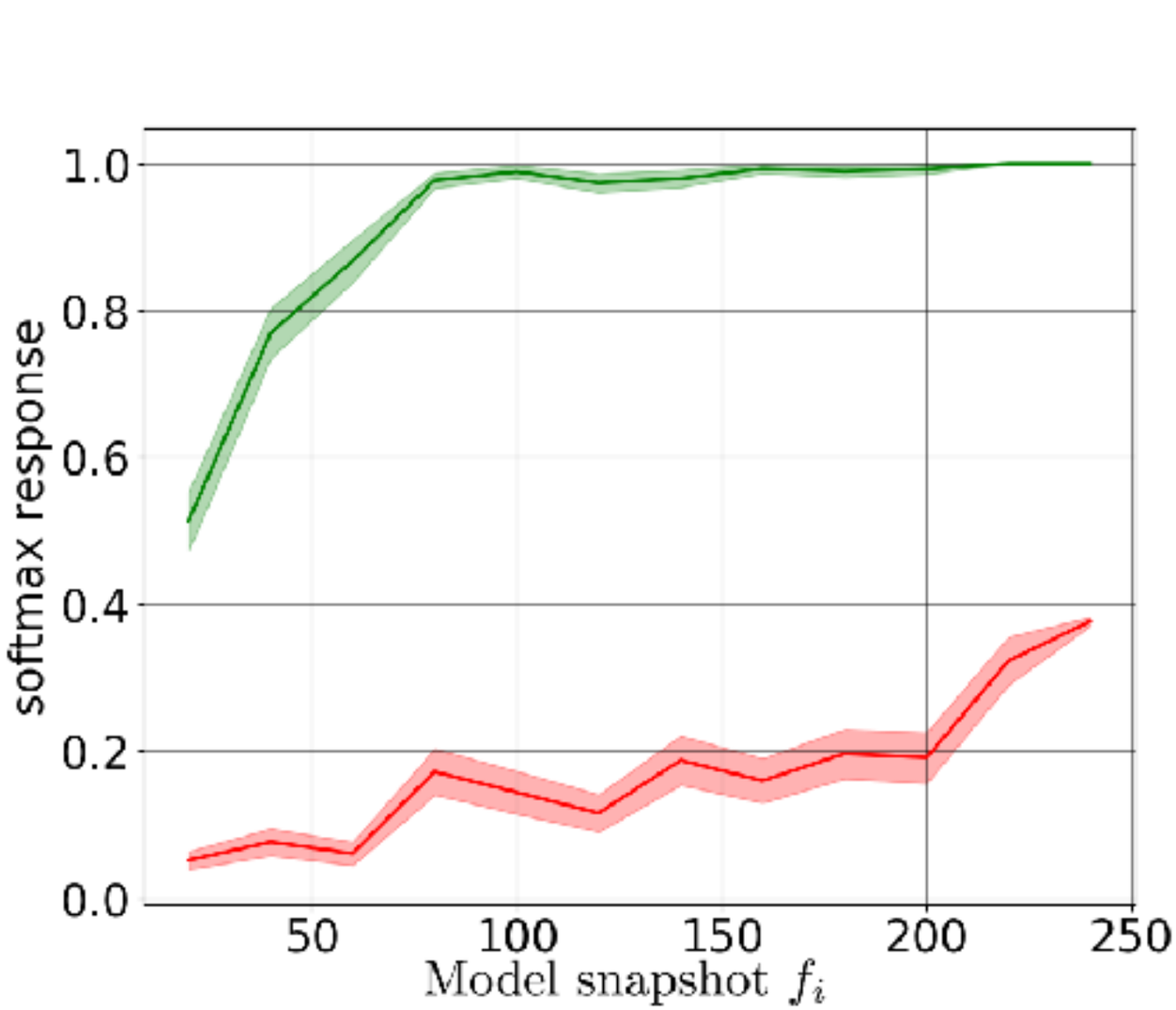
Bias-reduced Uncertainty

Non Bayesian uncertainty estimation

- Many optional confidence scores
- Our algorithm enhance any known confidence function
- It is based on two observations

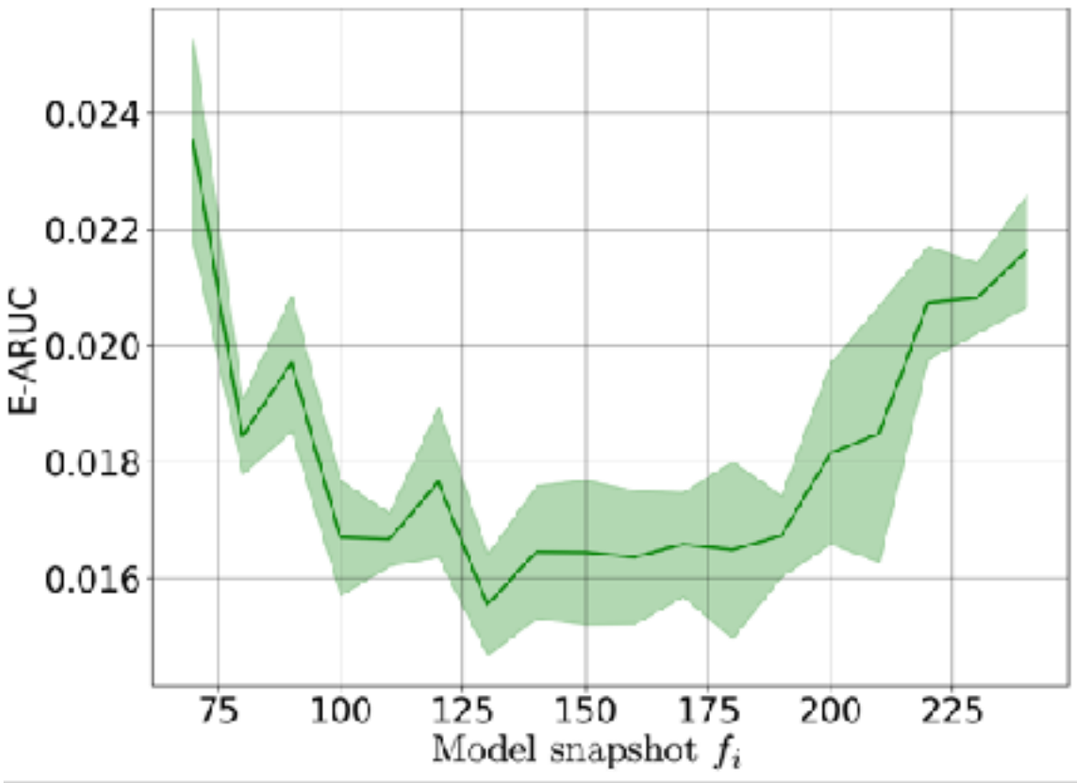
Observation 1 - Learning point

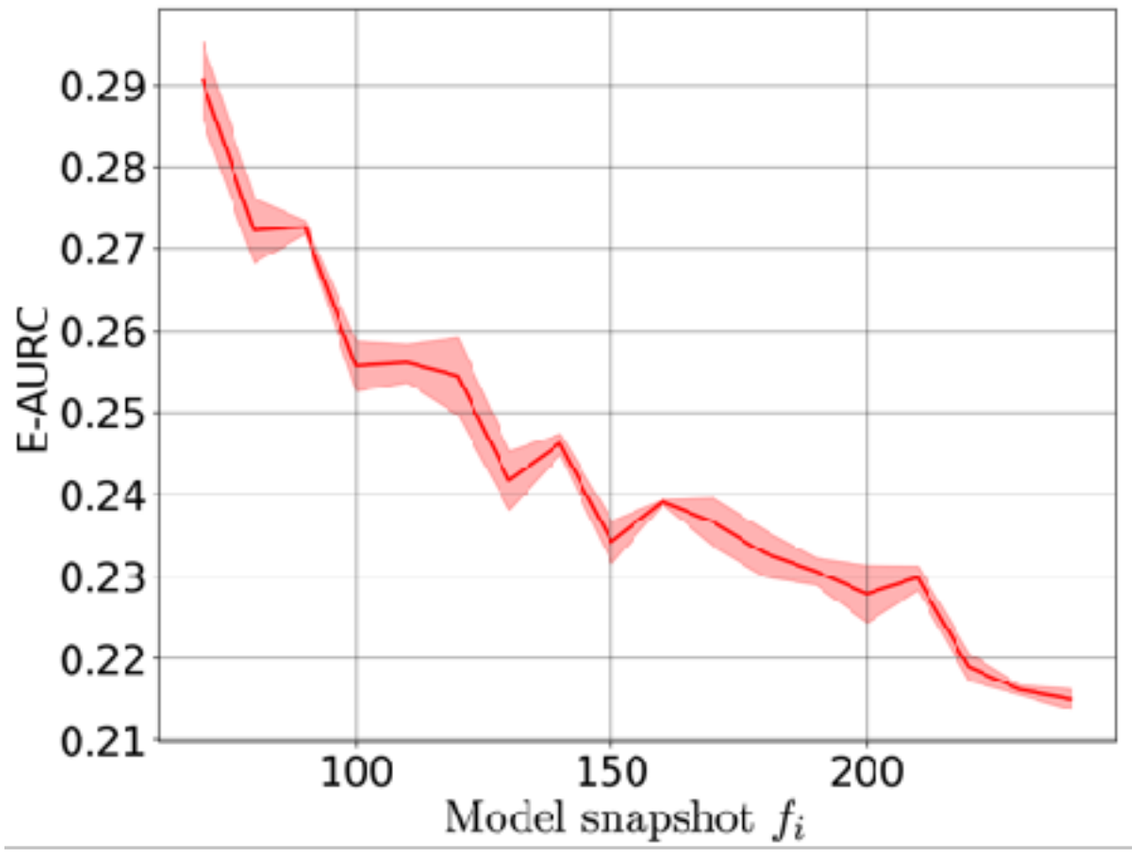
- We observed that “easy points” learned early during training.



Observation 2 - Overfitting

- We observed that uncertainty estimation for high confident points degrades at some point.





Solution - Early stopping

- Lets use information from intermediate points during training
- We propose two algorithms:
  - Supervised method - requires additional set
  - Approximated method

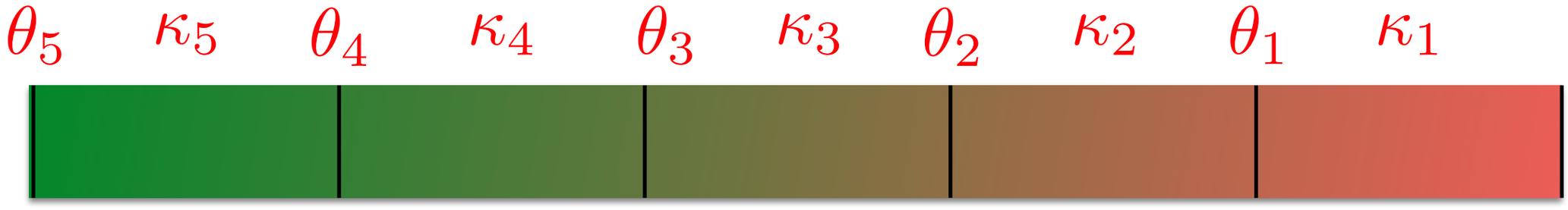
Point-wise Early Stopping (PES)

- Given an independent training set
  - We split the training set into confidence bins
    - For each bin we find the best intermediate model to predict uncertainty
- At test time:
  - we find the bin for a given point
  - We predict its uncertainty based on its bin.

Point-wise Early Stopping (PES)

- For each bin we search for the intermediate model that minimize the E-AURC for the points in the bin.

$$j = argmin_{0<j\leq T}(\text{E-AURC}(\kappa(x, \hat{y}_{f^{[T]}}(x)|f^{[j]}), f^{[T]}|S)$$



Averaged Early Stopping (AES)

- Simply average the confidence scores

$$F \triangleq \{f^{[i]} : i \in \mathbf{linspace}(t, T, k)\},$$

$$\kappa(x, \hat{y}_f(x)|F) \triangleq \frac{1}{k} \sum_{f^{[i]} \in F} \kappa(x, \hat{y}_f(x), f^{[i]})$$

Results PES

- We applied the PES over softmax response on four datasets

Dataset	E-AURC - SR	E-AURC - PES	% Improvement
CIFAR-10	4.6342 ± 0.07	$\mathbf{4.3543 \pm 0.06}$	**6.04**
CIFAR-100	51.3172 ± 0.43	$\mathbf{41.9579 \pm 0.39}$	**18.24**
SVHN	4.1534 ± 0.18	$\mathbf{3.7622 \pm 0.16}$	**9.41**
Imagenet	97.1393 ± 0.77	$\mathbf{94.8668 \pm 0.85}$	**2.34**

* E-AURC values multiplied by 1000 for clarity

Results AES

	Baseline	AES ($k = 10$)		AES ($k = 30$)		AES ($k = 50$)	
	E-AURC	E-AURC	%	E-AURC	%	E-AURC	%
CIFAR-10							
Softmax	4.78	4.81	-0.7	**4.49**	**6.1**	4.49	6.0
NN-distance	35.10	5.20	85.1	4.70	86.6	**4.58**	**86.9**
MC-dropout	5.03	5.32	-5.8	**4.99**	**0.9**	5.01	0.4
Ensemble	3.74	3.66	2.1	**3.50**	**6.5**	3.51	6.2
CIFAR-100							
Softmax	50.97	41.64	18.3	39.90	21.7	**39.68**	**22.1**
NN-distance	45.56	36.03	20.9	35.53	22.0	**35.36**	**22.4**
MC-dropout	47.68	49.45	-3.7	46.56	2.3	**46.50**	**2.5**
Ensemble	34.73	31.10	10.5	**30.72**	**11.5**	30.75	11.5
SVHN							
Softmax	4.24	**3.73**	**12.0**	3.77	11.1	3.73	11.9
NN-distance	10.08	**7.69**	**23.7**	7.81	22.5	7.75	23.1
MC-dropout	4.53	3.79	16.3	3.81	15.8	**3.79**	**16.3**
Ensemble	3.69	**3.51**	**4.8**	3.55	3.8	3.55	4.0
ImageNet							
Softmax	99.68	96.88	2.8	96.09	3.6	**94.77**	**4.9**
Ensemble	90.95	**88.70**	**2.47**	88.84	2.32	88.86	2.29

* E-AURC values multiplied by 1000 for clarity

Selective Classification

Knowledge

Knowledge

Known knowns

Known unknowns

Unknown unknowns

Selection with Guaranteed Risk (SGR)

- A selective classifier obtained by thresholding the confidence rate function

$$g_\theta(x) \triangleq \begin{cases} 1, & \text{if } \kappa_f(x) \geq \theta; \\ 0, & \text{otherwise.} \end{cases}$$

- Given a training set S_m , a desired risk r^* , and a confidence parameter δ , our goal is to learn a selective classifier such that:

$$Pr_{S_m}\{R(f,g) > r^*\} < \delta$$

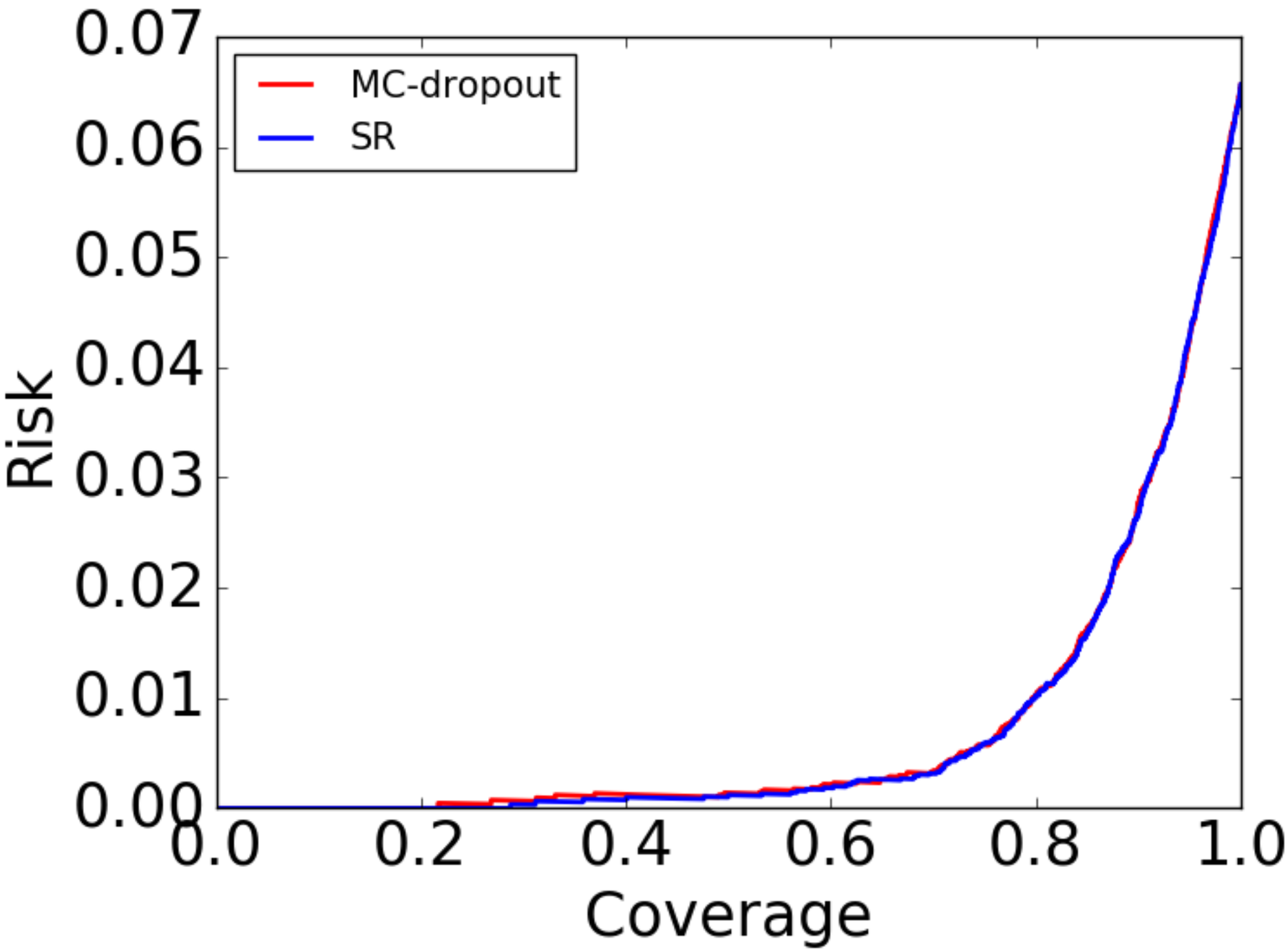
SGR Algorithm

- A generalization bound for DNNs
- The tightest bound possible
- Can work on a pre-trained network

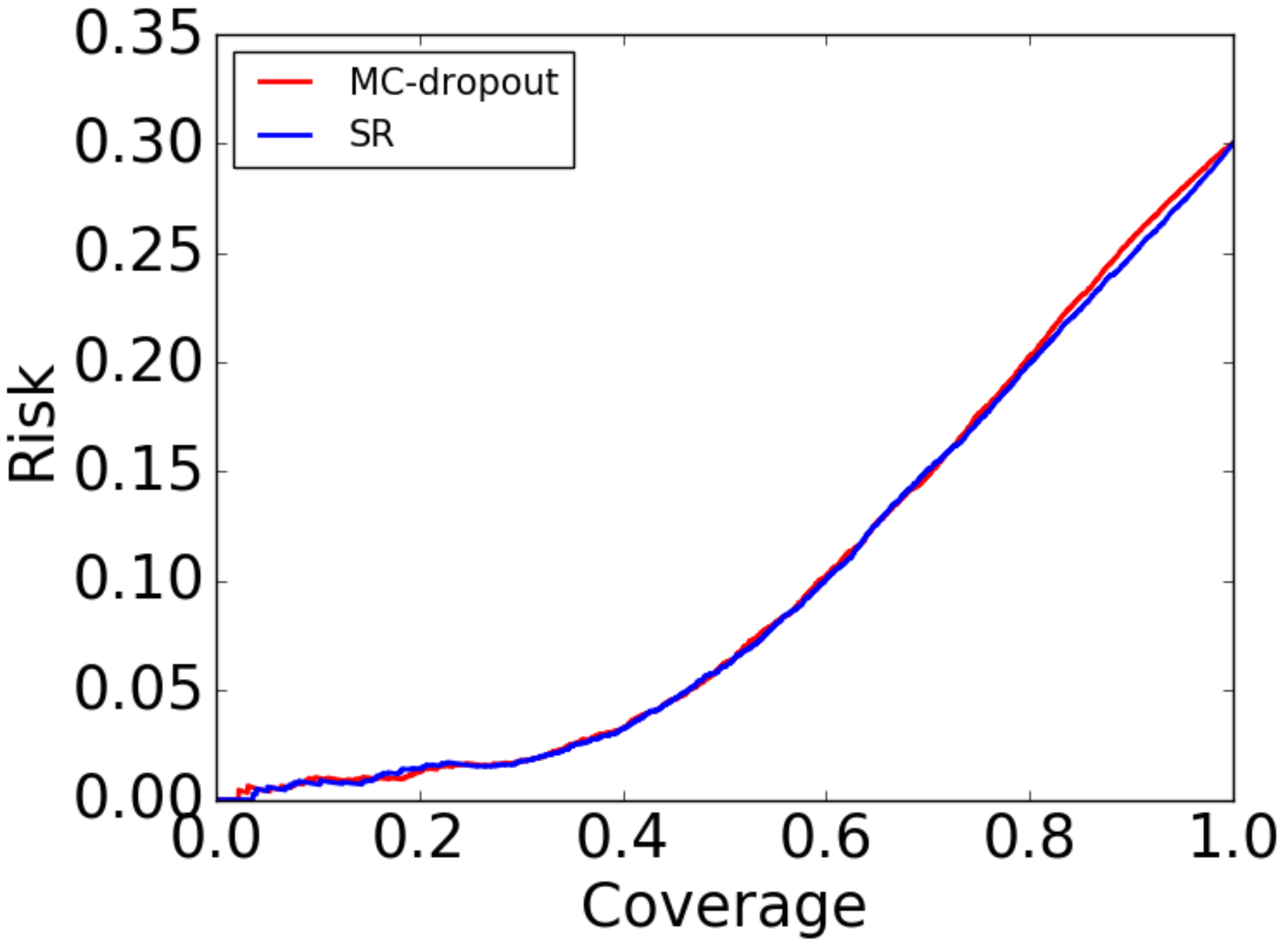
Experimental Setting

- Datasets:
  - CIFAR-10 - VGG-16
  - CIFAR-100 - VGG-16
  - IMAGENET - VGG-16 + Resnet-50 (top1 and top 5)

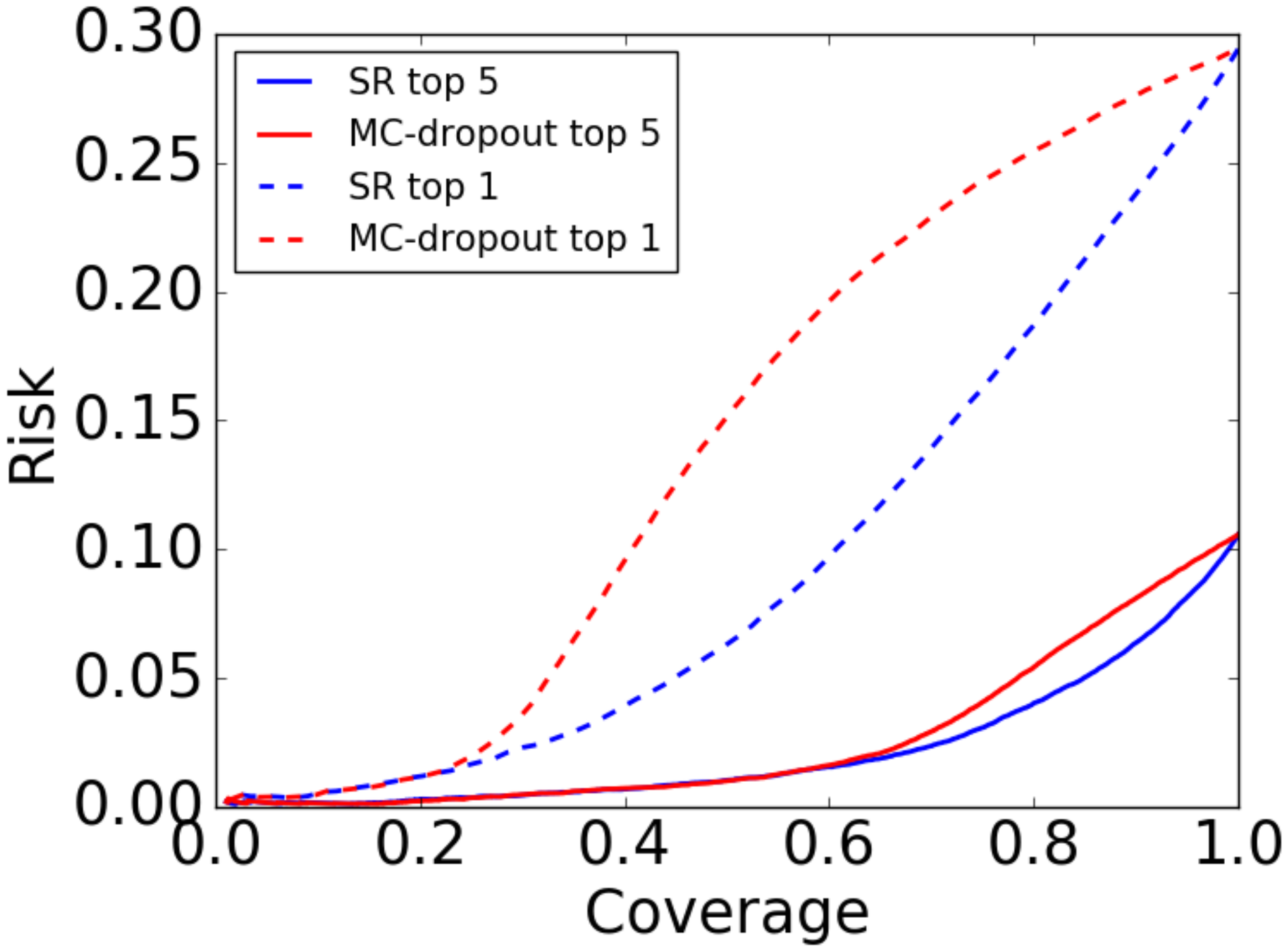
Experiments - RC-curve - CIFAR-10



Experiments - RC-curve - Cifar-100



Experiments - RC-curve Imagenet



Experiments - SGR

- CIFAR-10 - VGG-16

Desired risk (r^*)	Train risk	Train coverage	Test risk	Test coverage	Risk bound (b^*)
0.01	0.0079	0.7822	0.0092	0.7856	0.0099
0.02	0.0160	0.8482	0.0149	0.8466	0.0199
0.03	0.0260	0.8988	0.0261	0.8966	0.0298
0.04	0.0362	0.9348	0.0380	0.9318	0.0399
0.05	0.0454	0.9610	0.0486	0.9596	0.0491
0.06	0.0526	0.9778	0.0572	0.9784	0.0600

- IMAGENET - top 5 with Resnet-50

Desired risk (r^*)	Train risk	Train coverage	Test risk	Test coverage	Risk bound(b^*)
0.01	0.0080	0.3796	0.0085	0.3807	0.0099
0.02	0.0181	0.5938	0.0189	0.5935	0.0200
0.03	0.0281	0.7122	0.0273	0.7096	0.0300
0.04	0.0381	0.8180	0.0358	0.8158	0.0400
0.05	0.0481	0.8856	0.0464	0.8846	0.0500
0.06	0.0581	0.9256	0.0552	0.9231	0.0600
0.07	0.0663	0.9508	0.0629	0.9484	0.0700

Questions?

Papers

- Uncertainty:
    - Geifman, Yonatan, Guy Uziel, and Ran El-Yaniv. “**Bias-Reduced Uncertainty Estimation for Deep Neural Classifiers**." *arXiv preprint arXiv:1805.08206* (2018).
- Selective Classification:
    - Geifman, Yonatan, and Ran El-Yaniv. "**Selective classification for deep neural networks**." *Advances in neural information processing systems*. 2017.
- Active Learning:
    - Geifman, Yonatan, and Ran El-Yaniv. "**Deep Active Learning over the Long Tail.**" *arXiv preprint arXiv:1711.00941* (2017).
    - Geifman, Yonatan, and Ran El-Yaniv. "**Deep Active Learning with a Neural Architecture Search.**" *arXiv preprint arXiv:1811.07579* (2018).