

Uncertainty Estimation and its Applications in Deep Neural Networks

Yonatan Geifman
Advisor - Ran El-Yaniv

Motivation

- Machine Learning aims to solve several mission critical tasks
- Safe deployment of ML models requires:
 - Estimating of prediction uncertainty
 - Uncertainty control (rejection)

Work Outline

Uncertainty Estimation

Bias-Reduced Uncertainty Estimation
for Deep Neural Classifiers (ICLR 19)

Selective Classification

Selective classification for deep
neural networks (NeurIPS 17)

SelectiveNet: A Deep Neural
Network with an Integrated
Reject Option (ICML 19)

Active Learning

Deep active learning over the
long tail (preprint)

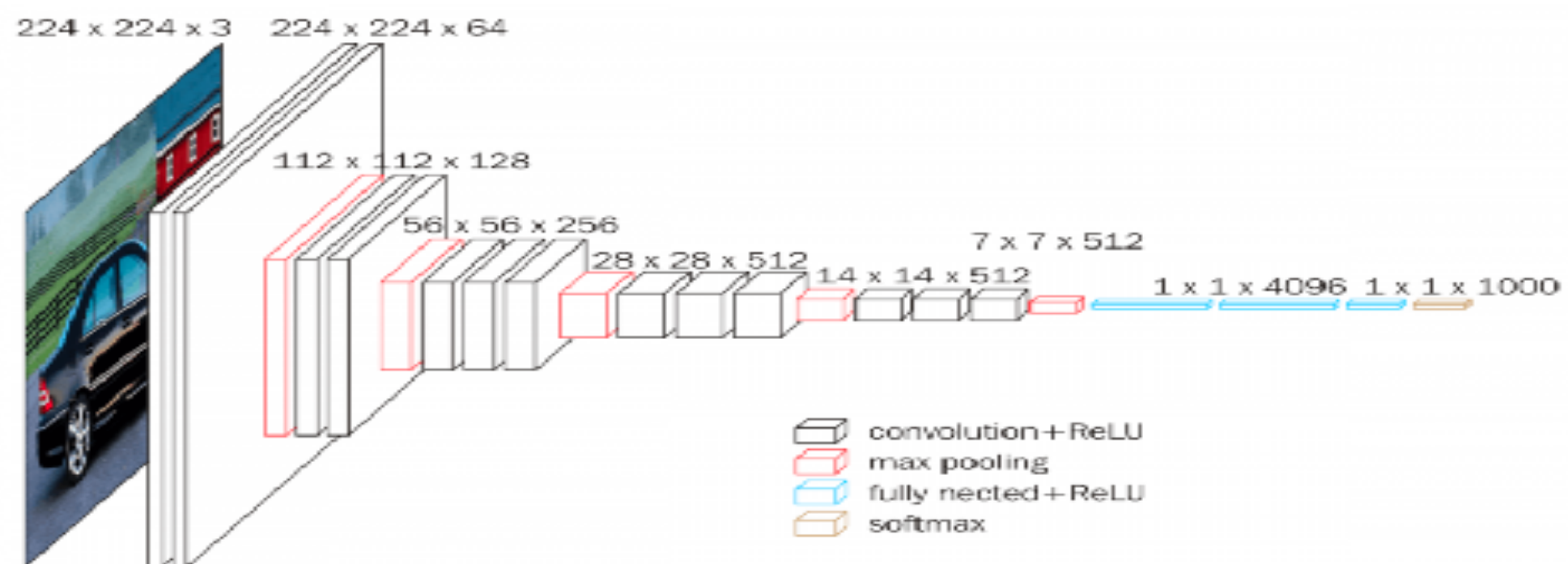
Deep active learning with a
neural architecture search
(under review)

Probability Calibration

How to Meaningfully Normalize
Any Loss Function (under review)

Deep Neural Networks

- Multiple layers of processing units
- The magic: Hierarchical feature representations learned by the network
- In this talk we focus on convolutional neural networks (CNNs)



Background - Uncertainty

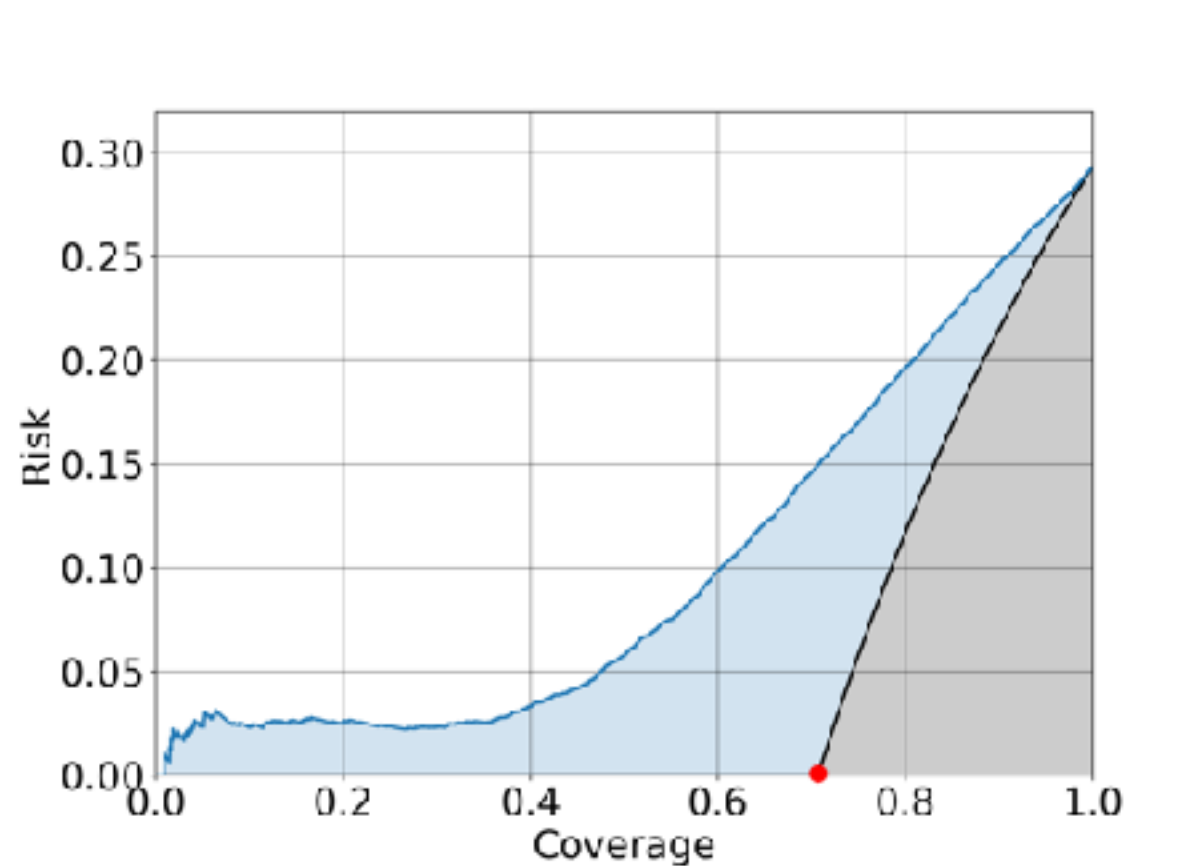
- Uncertainty has been studied since 1957 [Chow 1957]
- Selective Classification has been studied for various learning algorithms:
 - Support vector Machines [Wiener and El-Yaniv 2012]
 - Boosting [Cortes et al. 2016]
 - Nearest Neighbours [Hellman 1970]

Motivation - Deep Learning

- Large amount of activation information that can be transformed to uncertainty
 - Many layers
 - Many neurons in each layer
 - Each can be viewed as a classifier of a sub-task

Motivation - Deep Learning

- Far from being optimal - example Cifar-100



- For example - an application that requires 95% accuracy can reduce rejection rate by factor of 2

Motivation - Deep Learning

- Using “distance from decision boundary” works well for classification
- Regression is an open problem

Confidence Rate Functions

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

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

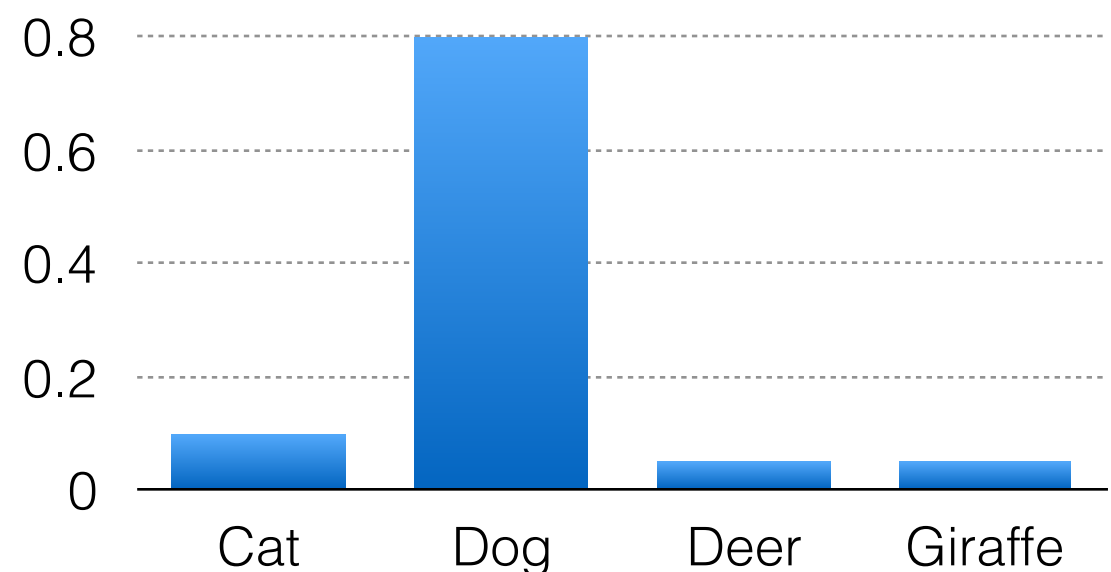
- We discuss three existing candidates:
 - Softmax response
 - MC-Dropout
 - Nearest neighbours distance

Confidence - Softmax Response

- Simply take κ to be the Softmax output

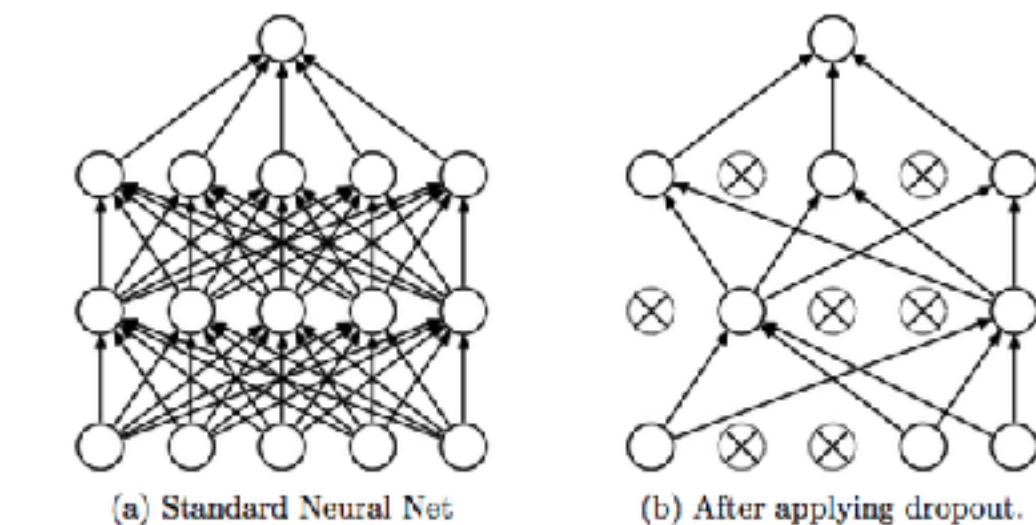
$$\kappa_f \triangleq \max_{j \in \mathcal{Y}} (f(x|j))$$

- Reflects the classification margin
- Other variants - Entropy, max-2nd activation



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

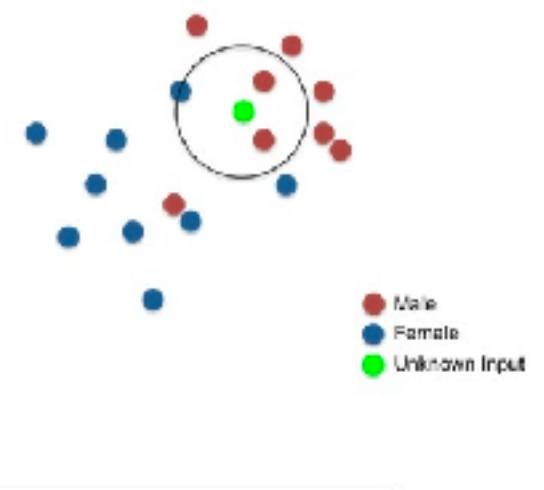


Gal, Yarin, and Zoubin Ghahramani. "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning."

Confidence - NN Distance

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

$$D(x) = \frac{\sum_{j=1, y^j = \hat{y}}^k e^{-\|f(x) - f(x_{train}^j)\|_2}}{\sum_{j=1}^k e^{-\|f(x) - f(x_{train}^j)\|_2}}$$



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

Statistical Learning

- Underlying unknown distribution $P(X, Y)$
- A labeled set $S_m = \{(x, y)\}^m \sim P$
- Our goal is to find $f \in \mathcal{F}$ that minimizes the risk:

$$R(f) \triangleq E_P[\ell(f(x), y)]$$

Selective Classification

- Selective Classifier is a pair (f, g)

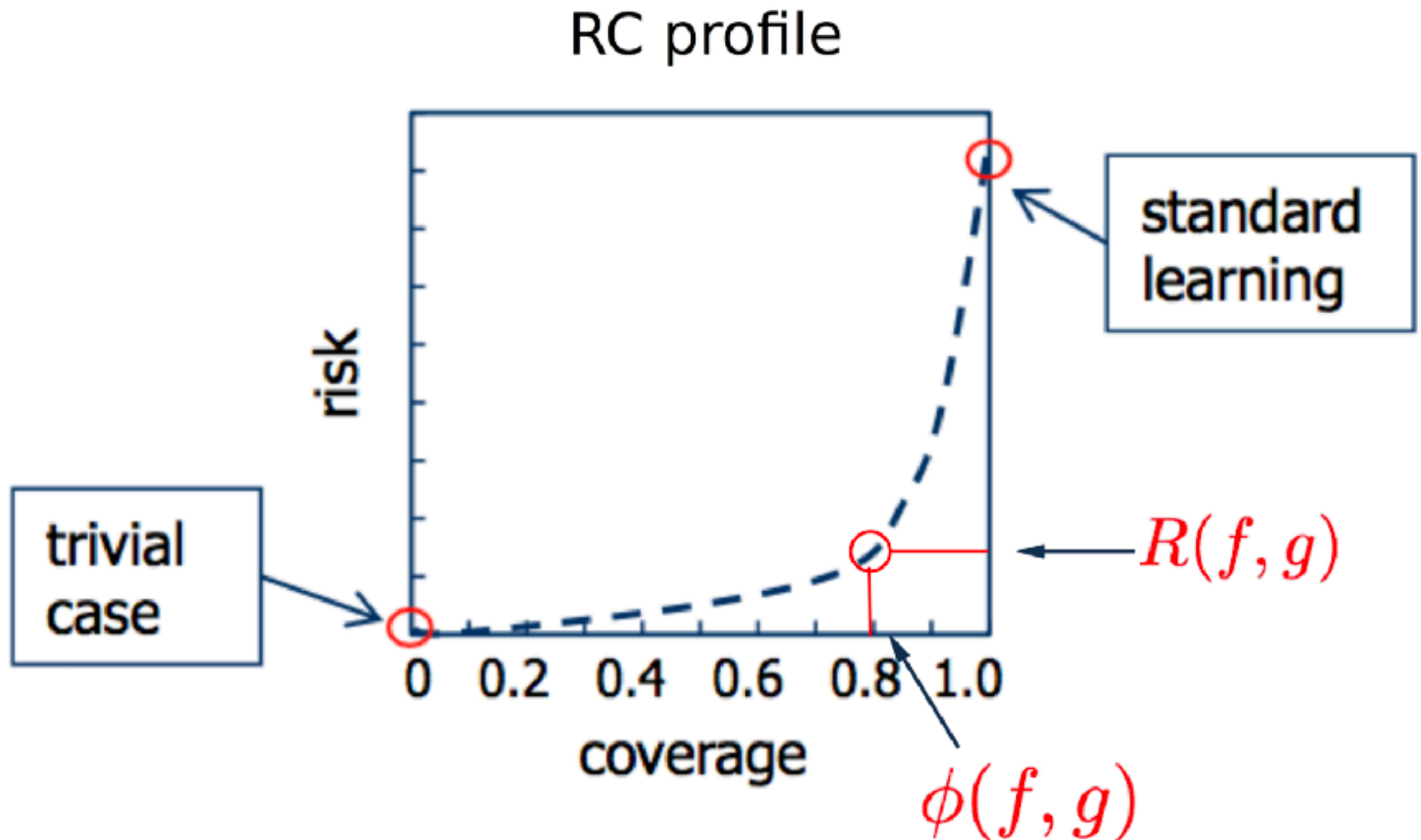
$$(f, g)(x) = \begin{cases} f(x), & \text{if } g(x) = 1; \\ \text{don't know}, & \text{if } g(x) = 0. \end{cases}$$

- Coverage:

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

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

Selective Classification



Selective Classification

Knowledge



Knowns



Unknowns

Knowledge

Known
knowns

Known
unknowns

Unknown
unknowns



“Reports that say that something hasn't happened are always interesting to me, because as we know, there are **known knowns**; there are things we know we know. We also know there are **known unknowns**; that is to say we know there are some things we do not know. But there are also **unknown unknowns**—the ones we don't know we don't know. And if one looks throughout the history of our country and other free countries, it is **the latter category that tend to be the difficult ones.**” Donald Rumsfeld

From Uncertainty to Selective Classifier

- A selective classifier can be 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 .

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 , the SGR algorithm find a selective classifier such that:

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

Lemma 1 - Binomial Tail

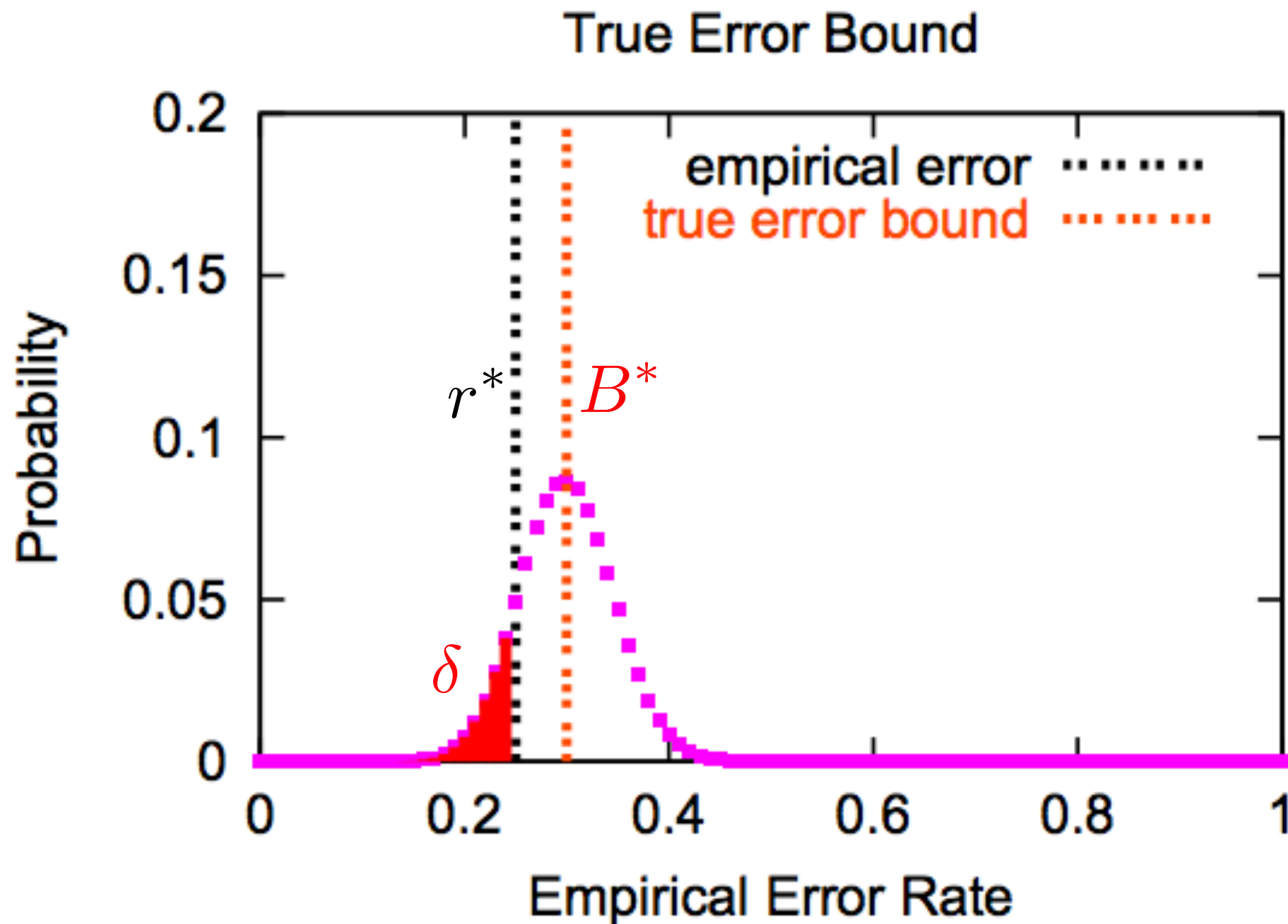
- Let $B^*(\hat{r}_i, \delta, S_m)$ be the solution b of the following equation

$$\sum_{j=0}^{m \cdot \hat{r}(f|S_m)} \binom{m}{j} b^j (1-b)^{m-j} = \delta.$$

Then

$$Pr_{S_m} \{R(f|P) > B^*(\hat{r}_i, \delta, S_m)\} < \delta$$

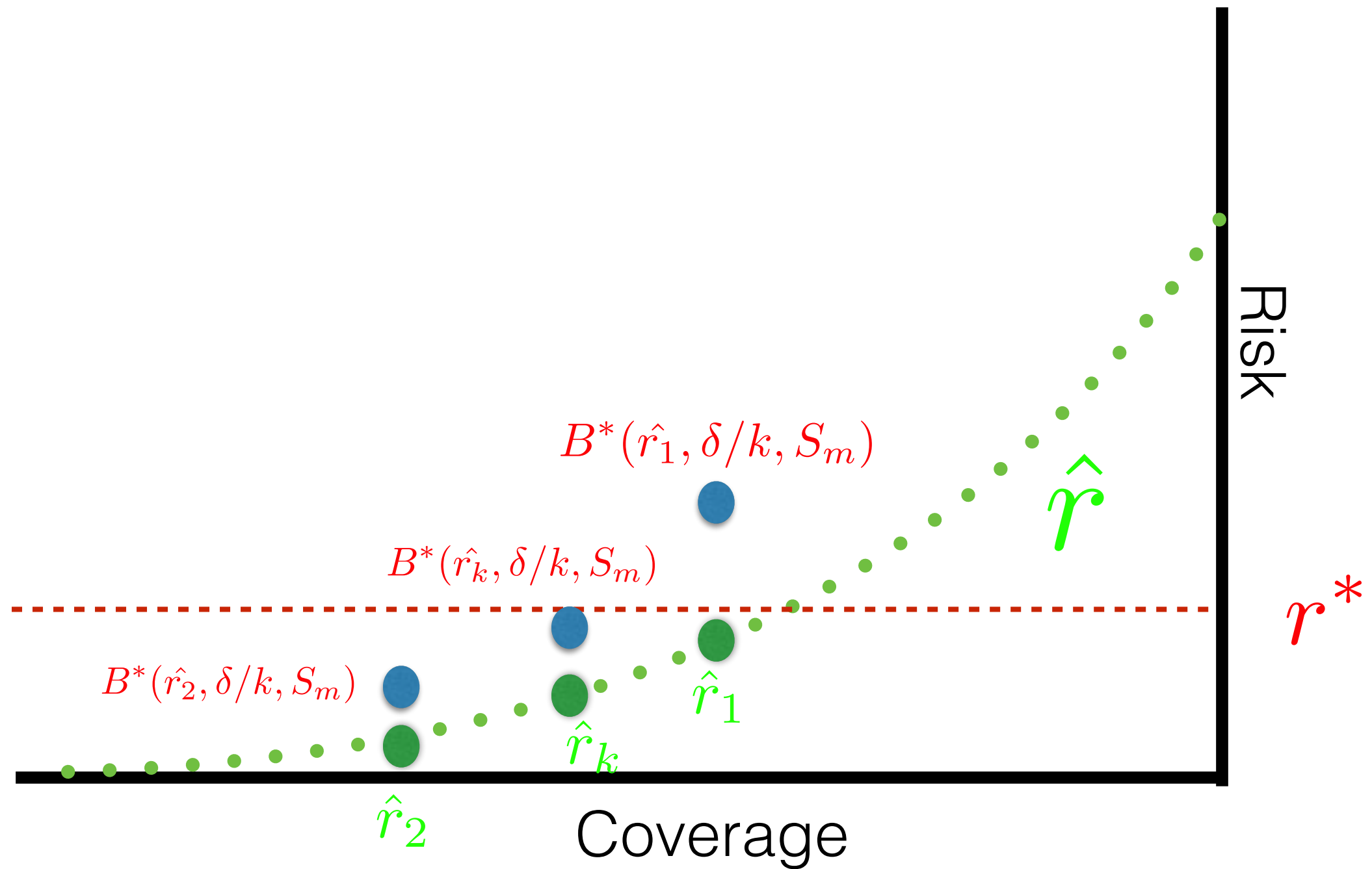
Lemma 1 - Binomial Tail



SGR Algorithm

- For a given training set $S_m \sim P(X, Y)$, a desired risk r^* and a confidence parameter δ
- set $k = \lceil \log(m) \rceil$
- Use binary search to find $\hat{\theta} \in \{\kappa(x) : x \in S_m\}$ such that $B^*(\hat{r}_\theta, \delta/k, S_m) \leq r^*$

SGR Algorithm



Theorem 1 - SGR Generalization bound

Theorem: For an application of SGR on $S_m \sim P(X, Y)$ with a given r^* and δ , the output (f, g_k) Satisfies

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

Theorem 1 - SGR Generalization bound - Proof Sketch

- On each iteration

$$Pr_{S_m} \{R(f, g_i) > B^*(\hat{r}_i, \delta, S_m)\} < \delta/k$$

- Due to the binary search

$$\exists i : B^*(\hat{r}_i, \delta, S_m) \leq r^*$$

- An application of the union bound among iterations complete the proof

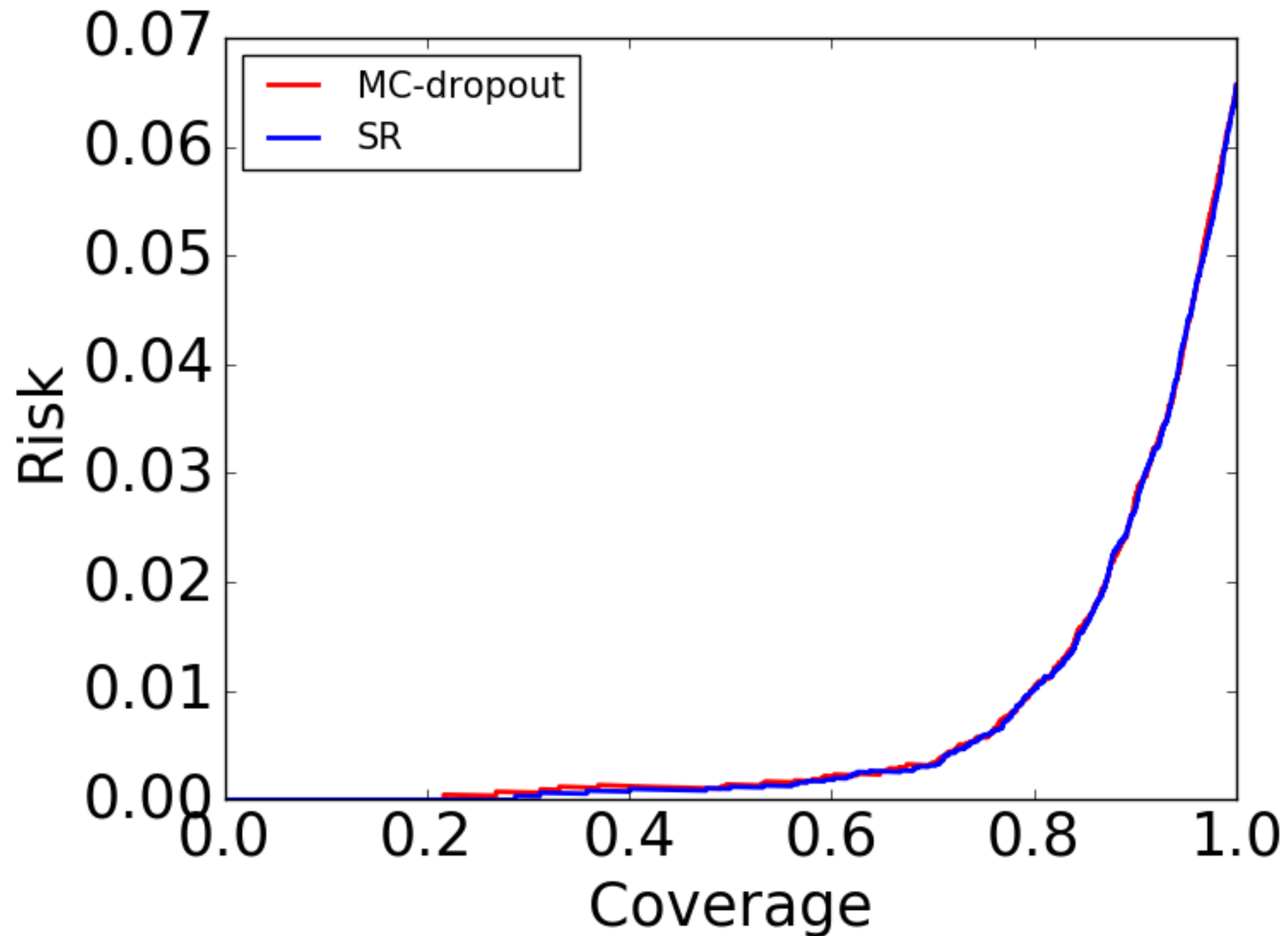
SGR Algorithm

- A generalization bound for DNNs
- The tightest bound possible (without other assumptions)
- Can be applied on a pre-trained network

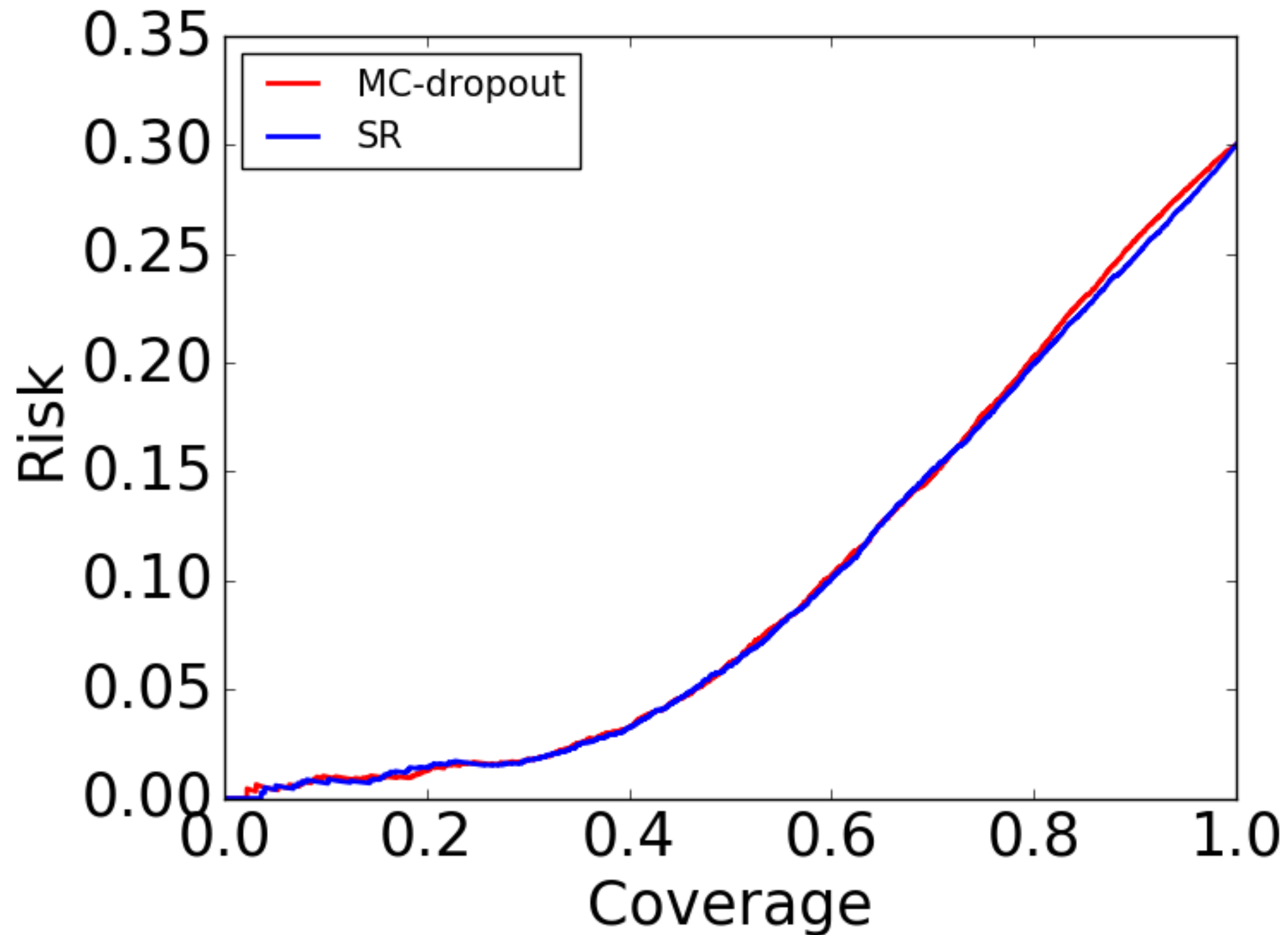
Experimental Setting

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

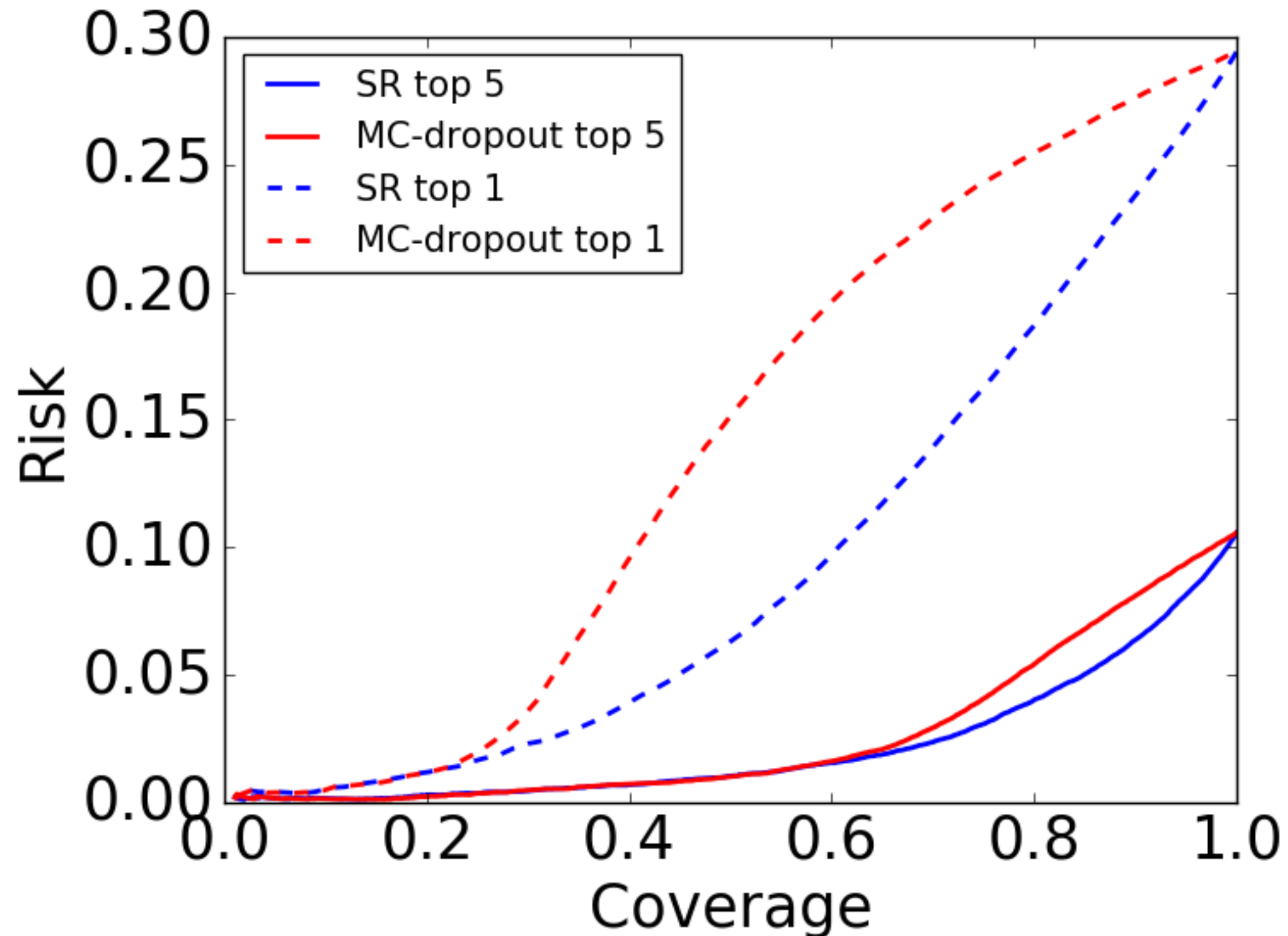
Experiments - RC-curve - CIFAR-10



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

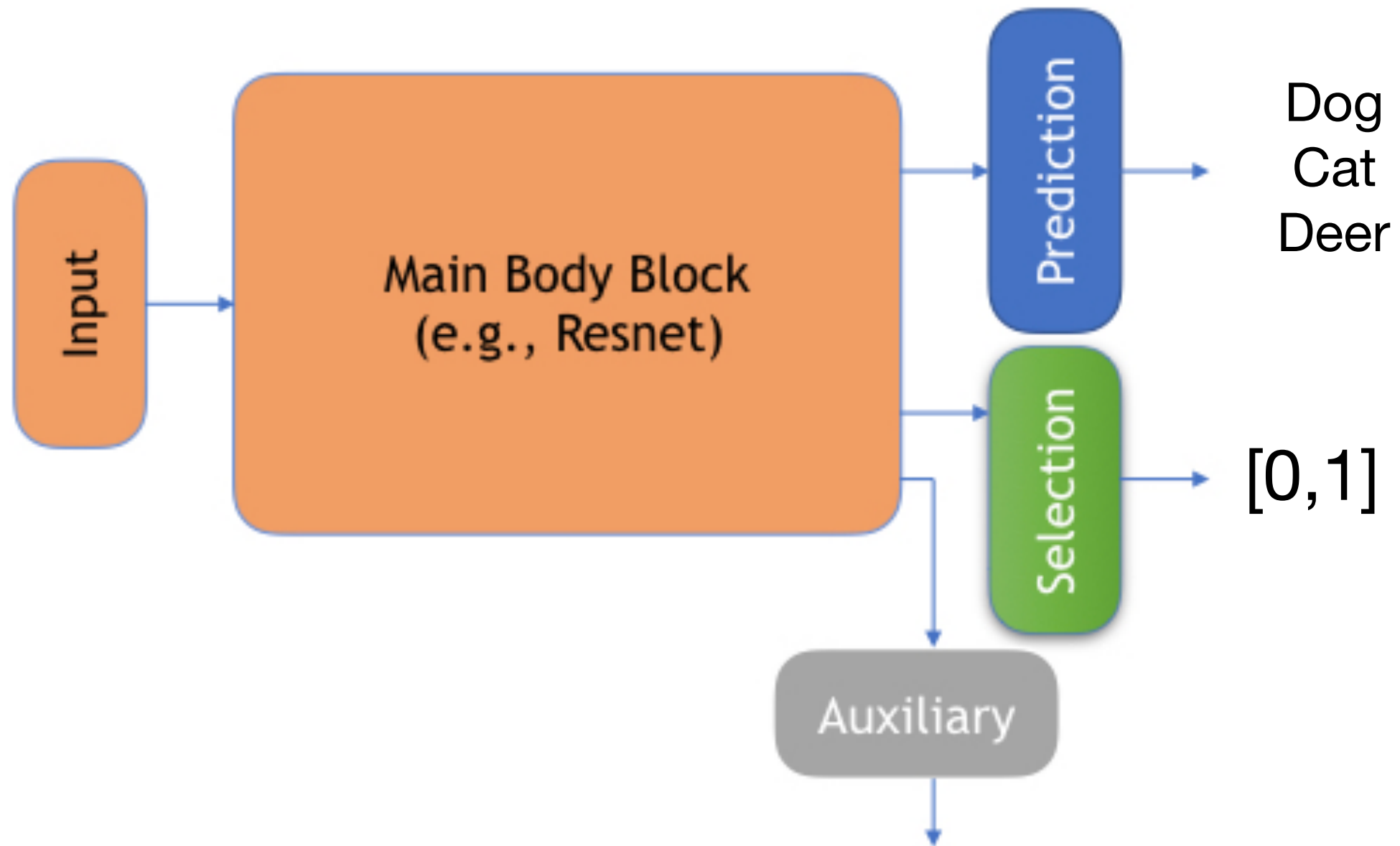
SelectiveNet

Background

- We saw how to transform a classifier to be a selective classifier using threshold over prediction uncertainty
- Can there be better uncertainty estimation
- Motivation - Test with 5/10 questions
- Joint optimization:

$$\begin{aligned}\theta^* &= \arg \min_{\theta \in \Theta} (R(f_\theta, g_\theta)) \\ &\quad s.t. \phi(g_\theta) \geq c.\end{aligned}$$

SelectiveNet



Optimization

- Inspired by interior point methods (IPM)
- Constrained optimization problem:

$$\begin{aligned}\theta^* &= \arg \min_{\theta \in \Theta} (R(f_\theta, g_\theta)) \\ &s.t. \ \phi(g_\theta) \geq c.\end{aligned}$$

- Unconstrained objective:

$$\mathcal{L}_{(f,g)} = \hat{r}_\ell(f, g|S_m) + \lambda \Psi(c - \hat{\phi}(g|S_m))$$

$$\Psi(a) = \max(0, a)^2$$

$$\hat{r}_\ell(f, g|S_m) = \frac{\frac{1}{m} \sum_{i=1}^m \ell(f(x_i), y_i) g(x_i)}{\hat{\phi}(g|S_m)}$$

Auxiliary output

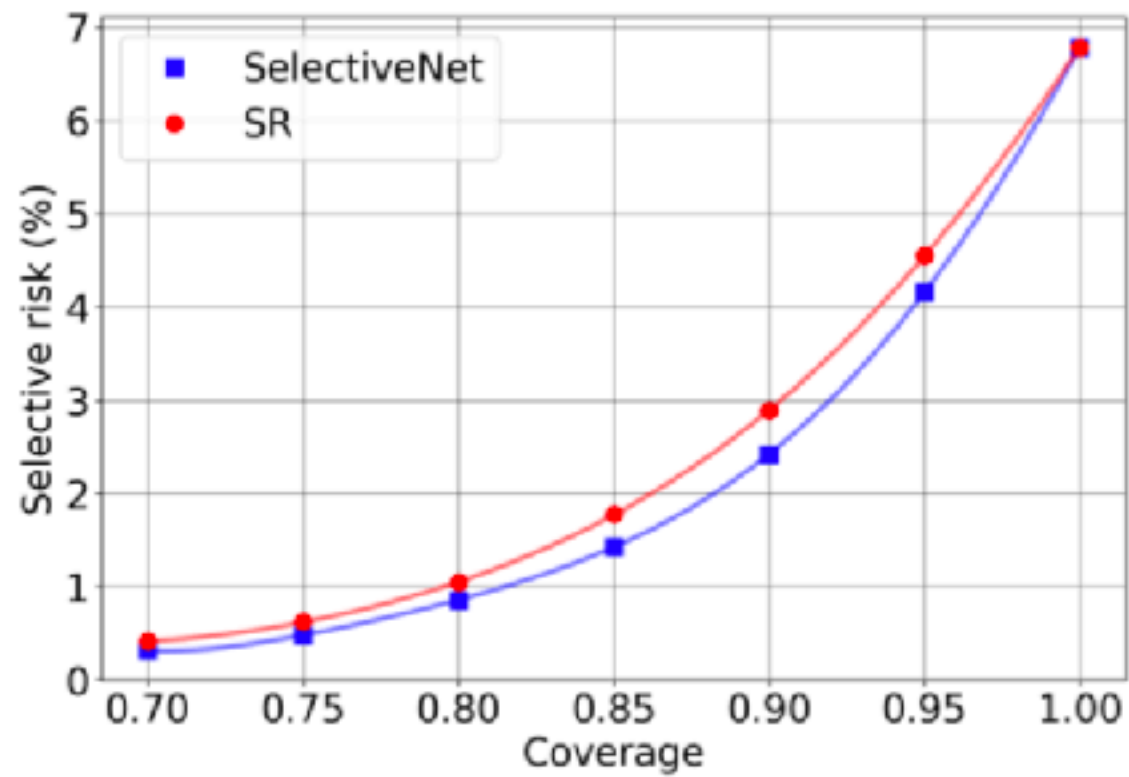
- In low coverage rate the effective training set size is reduced
- Auxiliary output is added as regularisation for representation learning based on all points

$$\mathcal{L}_h = \hat{r}(h|S_m) = \frac{1}{m} \sum_{i=1}^m \ell(h(x_i), y_i)$$

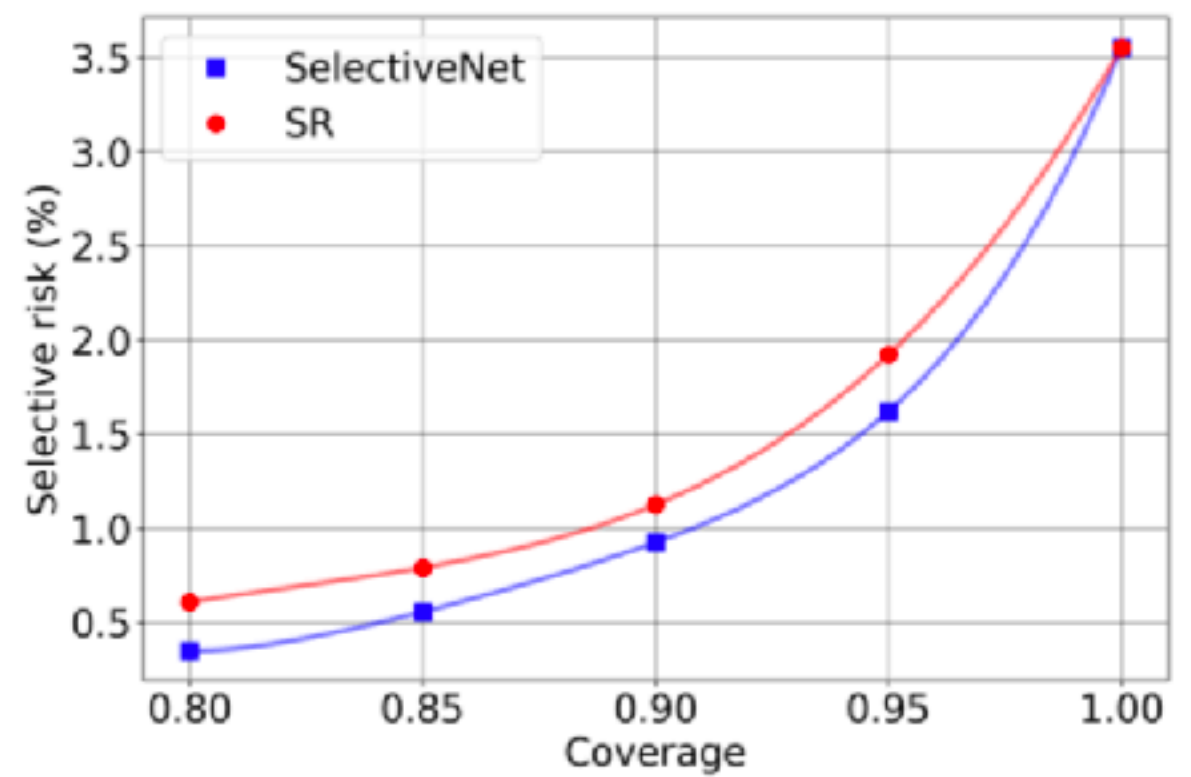
- Combined with the selective risk:

$$\mathcal{L} = \alpha \mathcal{L}_{(f,g)} + (1 - \alpha) \mathcal{L}_h$$

Empirical Results

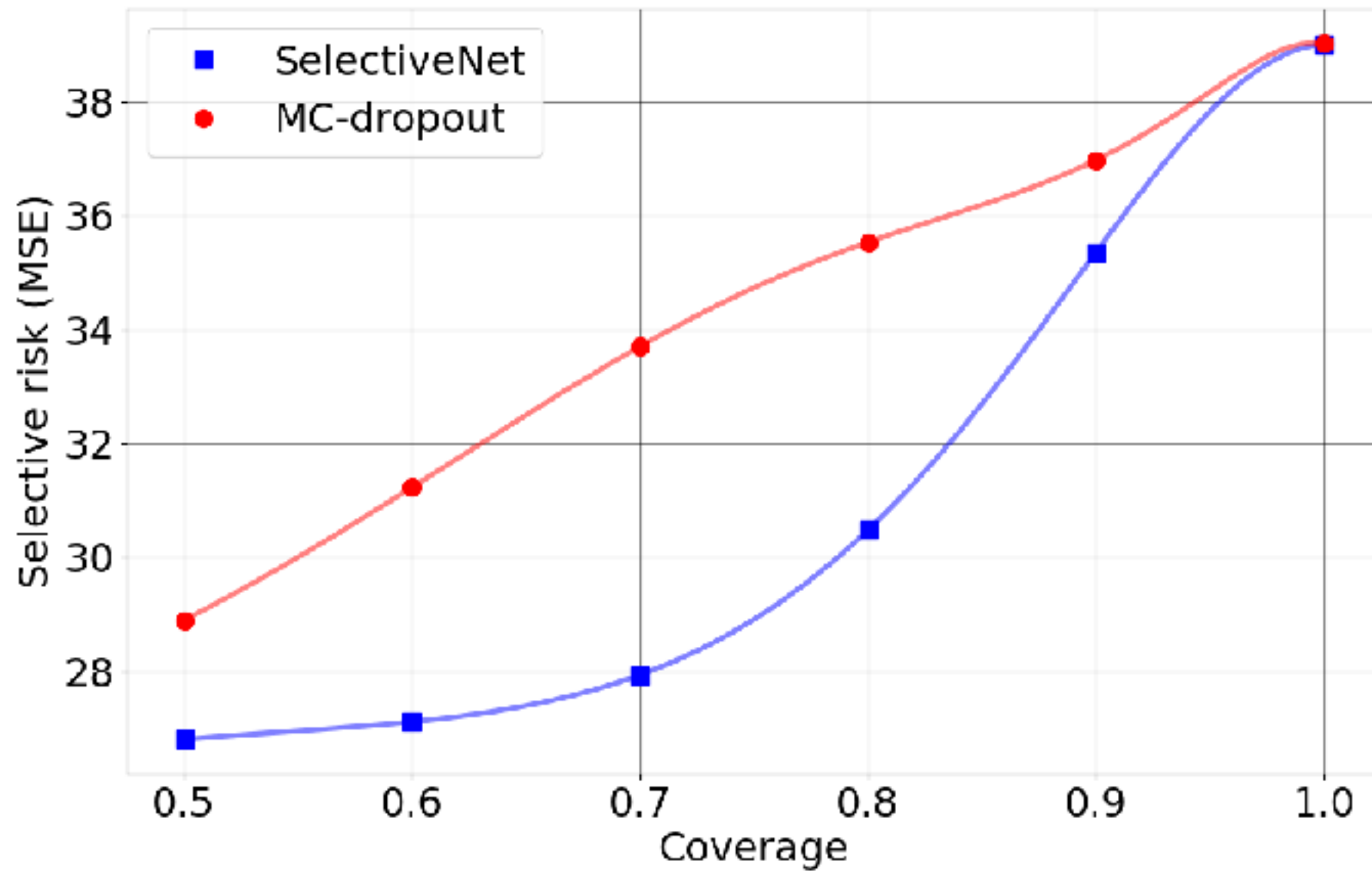


(a) Cifar-10

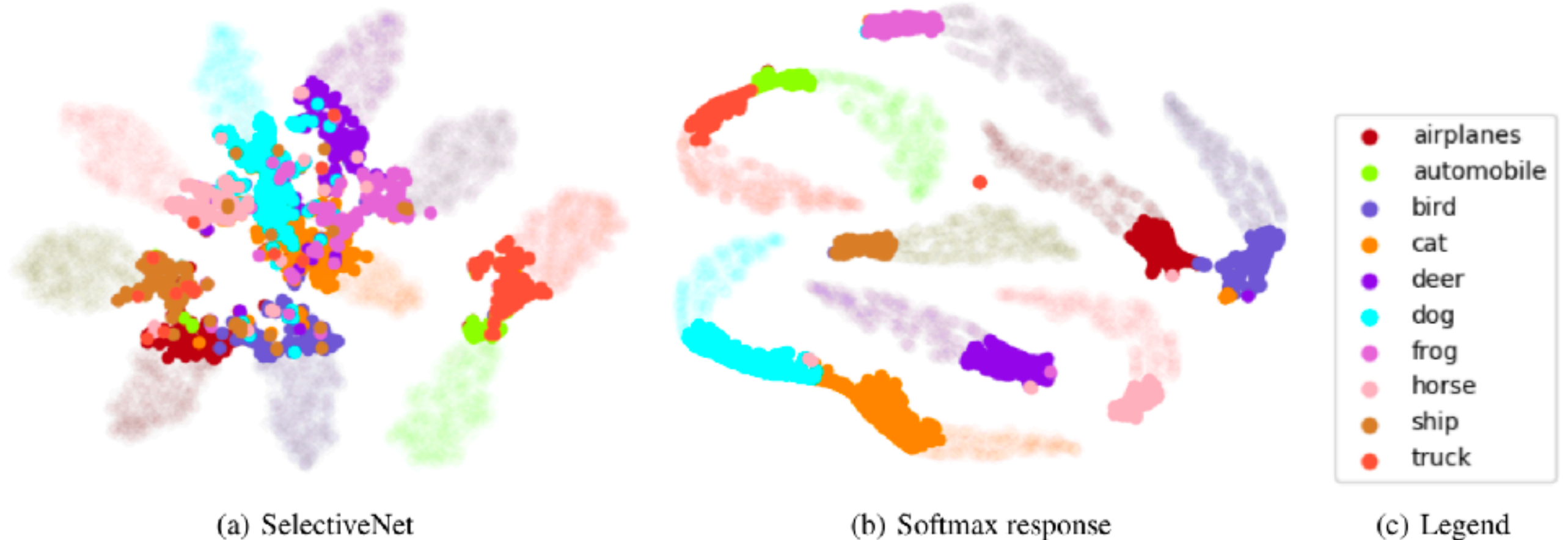


(b) Cats vs. dogs

Empirical Results - Regression















Embedding Analysis



- SelectiveNet does not “invest” representational capacity on rejected instances

Thresholds VS SelectiveNet

	Thresholds (MC-dropout, SR)	SelectiveNet
SOTA results		
Regression	 (Costly)	
Pretrained Network		
No additional training set		
Risk Control		 (Costly)
Various coverage rates		 (Costly)

Questions?

Publications and work in progress

- Uncertainty:
 - Geifman, Yonatan, Guy Uziel, and Ran El-Yaniv. "**Bias-Reduced Uncertainty Estimation for Deep Neural Classifiers.**" *International conference on Learning Representation (ICLR 2019)*
- Selective Classification:
 - Geifman, Yonatan, and Ran El-Yaniv. "**Selective classification for deep neural networks.**" *Advances in neural information processing systems*. 2017.
 - Geifman, Yonatan, and Ran El-Yaniv. "**SelectiveNet: A Deep Neural Network with an Integrated Reject Option**" *International Conference on Machine Learning (ICML 2019)*
- 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.**" *Under review*.
- Performance evaluation
 - Ran El-Yaniv, Yonatan Geifman*, Yair Wiener. "**How to Meaningfully Normalize any Loss Function**" under review