Continual Machine Learning
Summer 2023

Teacher

Dr. Martin Mundt,

hessian.AI-DEPTH junior research group leader on Open World Lifelong Learning (OWLL)
& researcher in the Artificial Intelligence and Machine Learning (AIML) group at TU Darmstadt

Time

Every Friday 14:25 - 16:05 CEST

Course Homepage

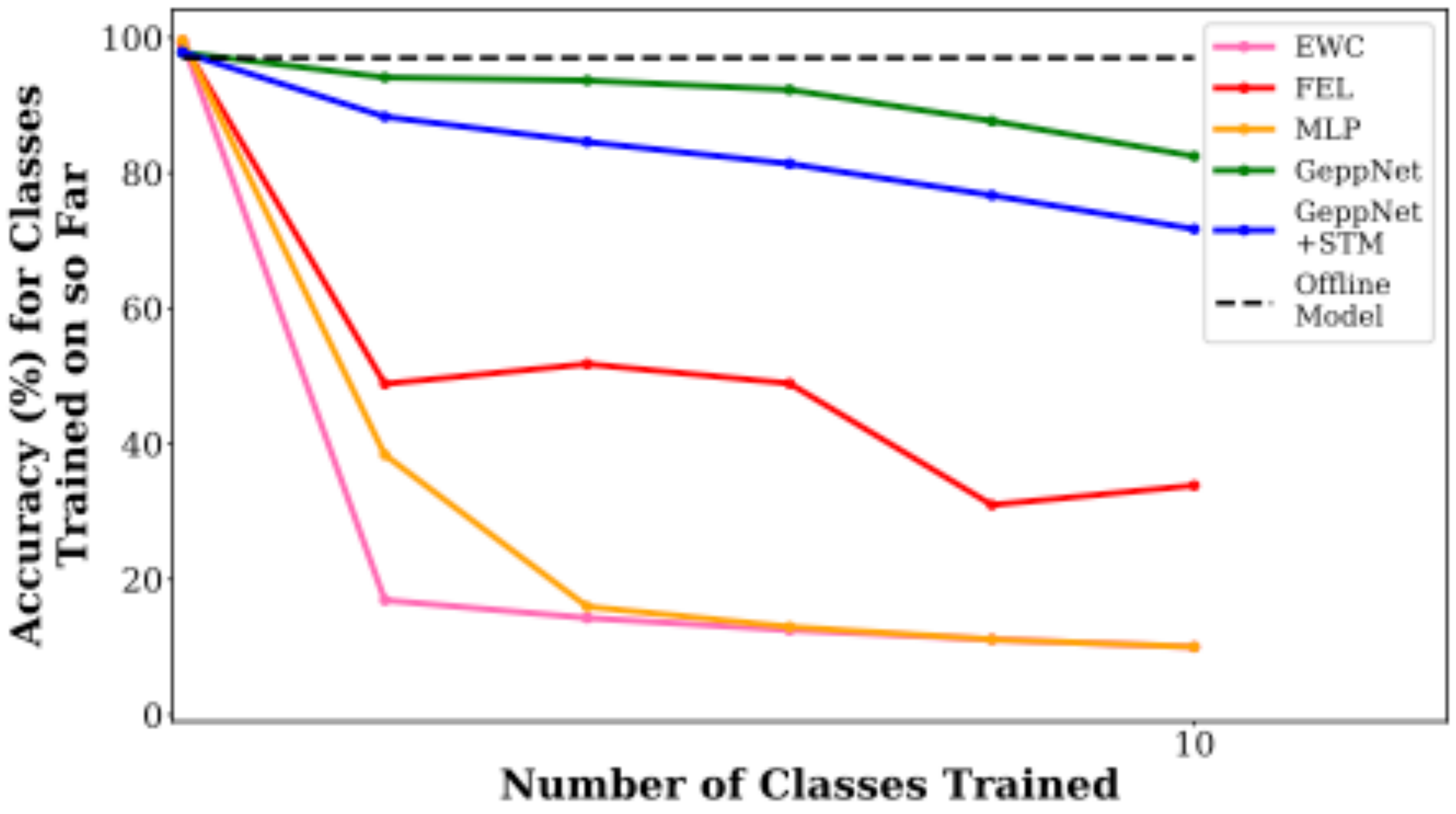
http://owll-lab.com/teaching/cl_lecture_23

https://www.youtube.com/playlist?list=PLm6QXeaB-XkA5-lVBB-h7XeYzFzgSh6sk

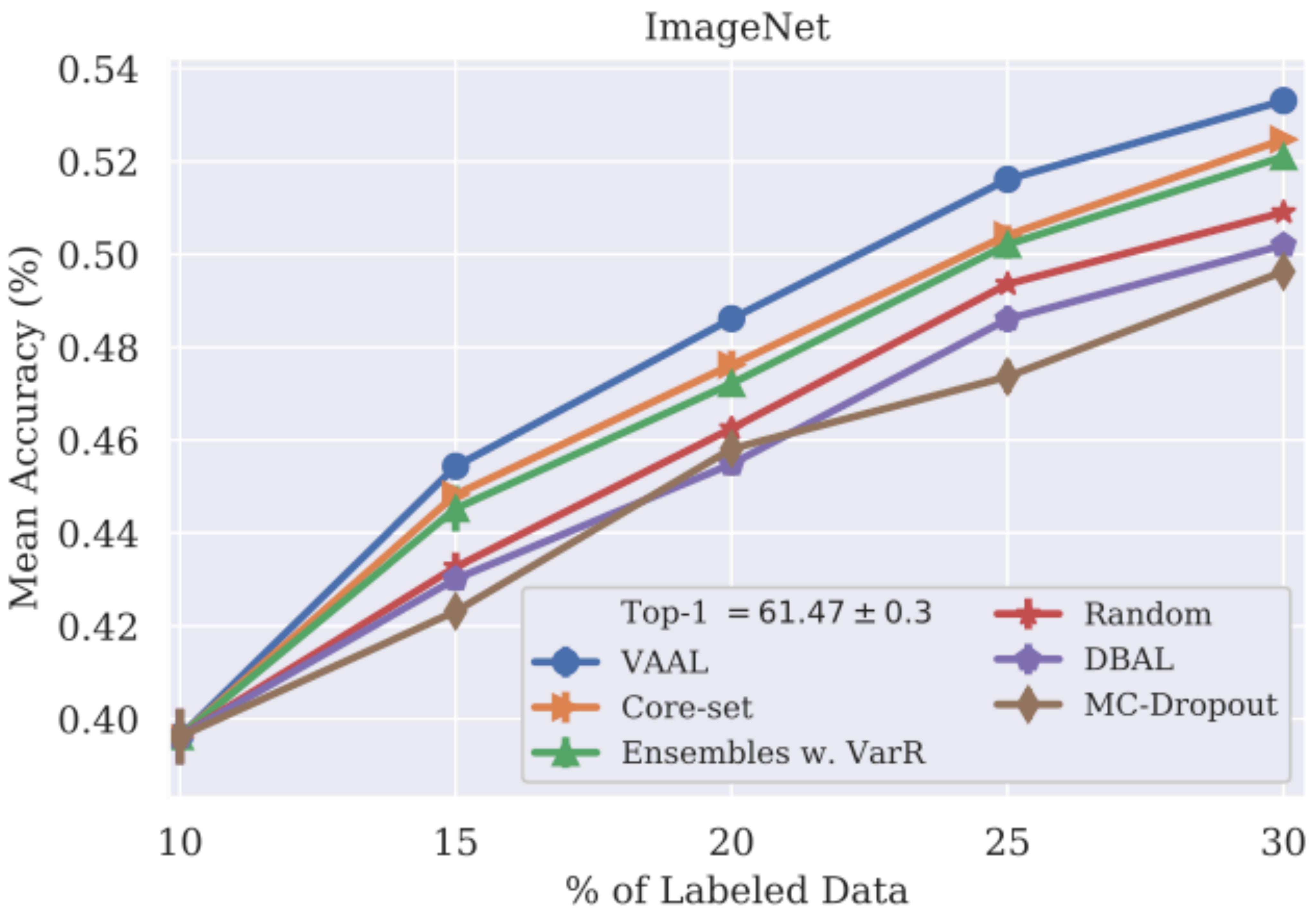
Week 8: Open world learning - learning & prediction in the presence of the unknown

Recall sequences so far

We've discussed various ways to measure + assumptions, but so far it was always clear what to test on



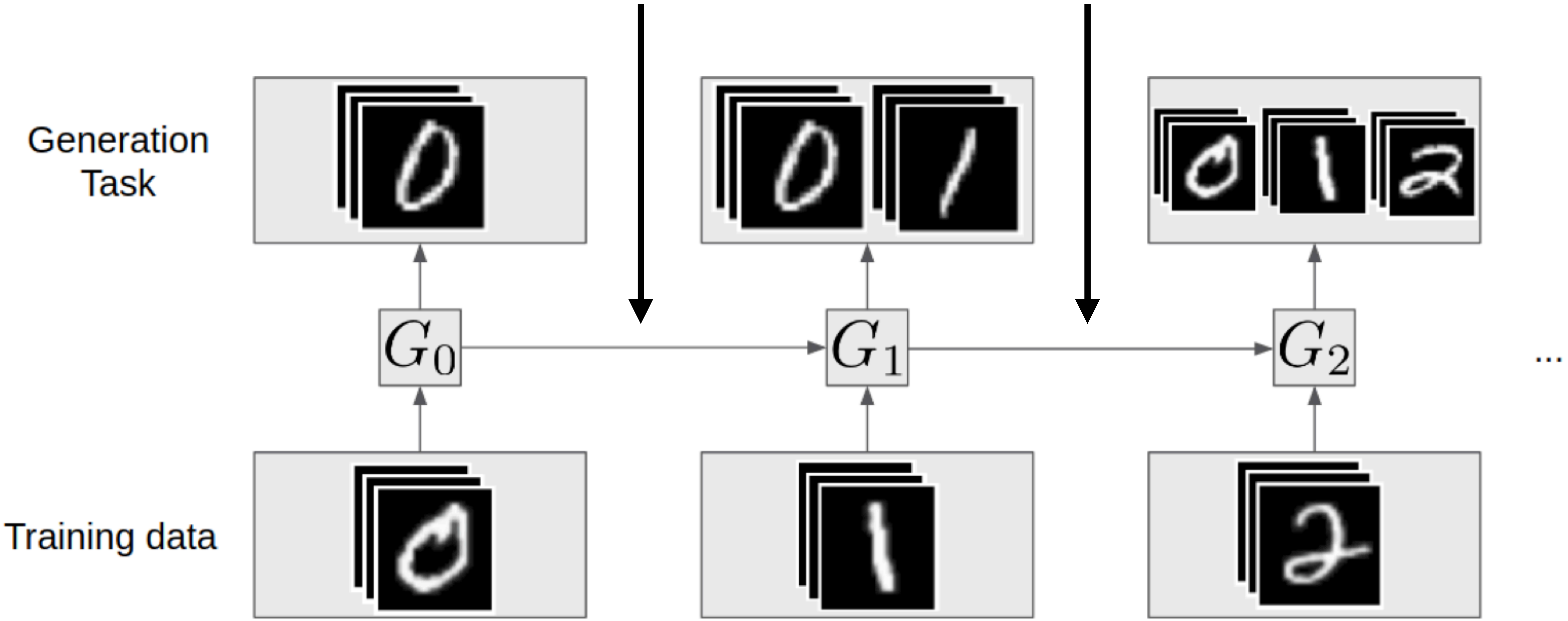
Kemker et al, "Measuring Catastrophic Forgetting in Neural Networks", AAAI 2018



Sinha et al, "Variational Adversarial Active Learning", ICCV 2019

Recall: the tasks we considered

What if we don't know the boundary & aren't constrained on our testing examples?



Lesort et al, "Generative Models from the perspective of Continual Learning", IJCNN 2019

Recall: the tasks we considered

What if we don't know the boundary & aren't constrained on our testing examples?

What if future or unrelated data is in the test set?

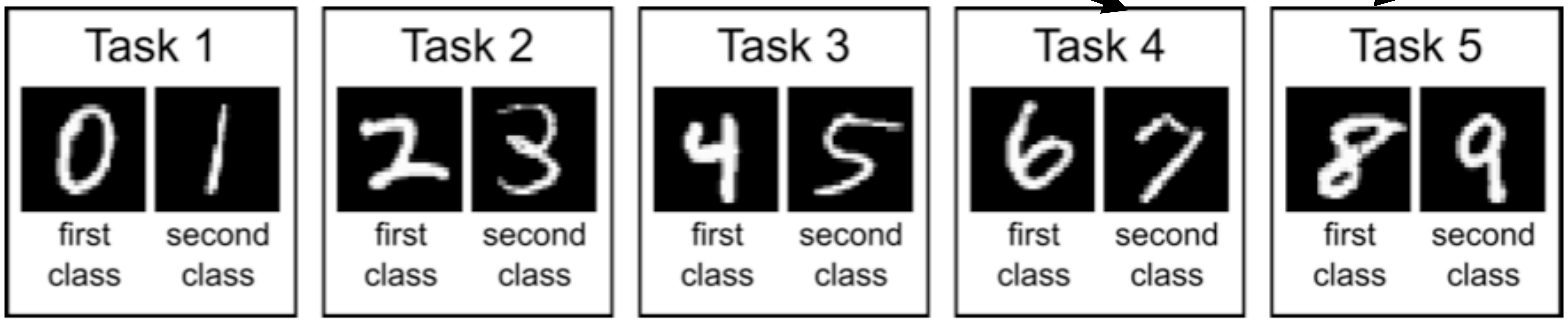
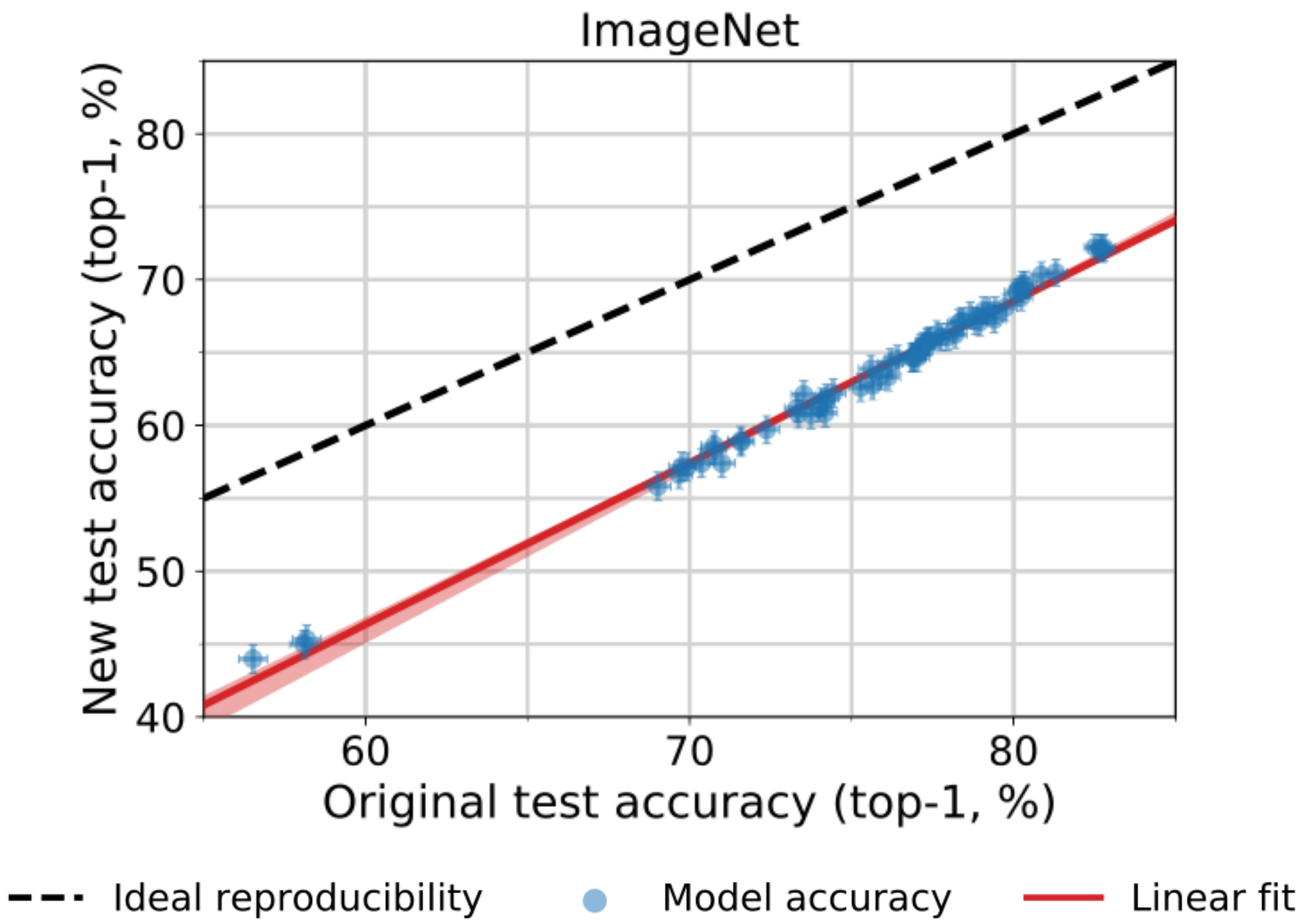


Figure 1: Schematic of split MNIST task protocol.

van de Ven et al, "Three types of incremental learning", Nature MI 2022

Recall: distribution shifts



Recall: natural data distributions are complex & can easily shift!

Performance loss even happens if we recollect another “test set” with the same instructions a second time!

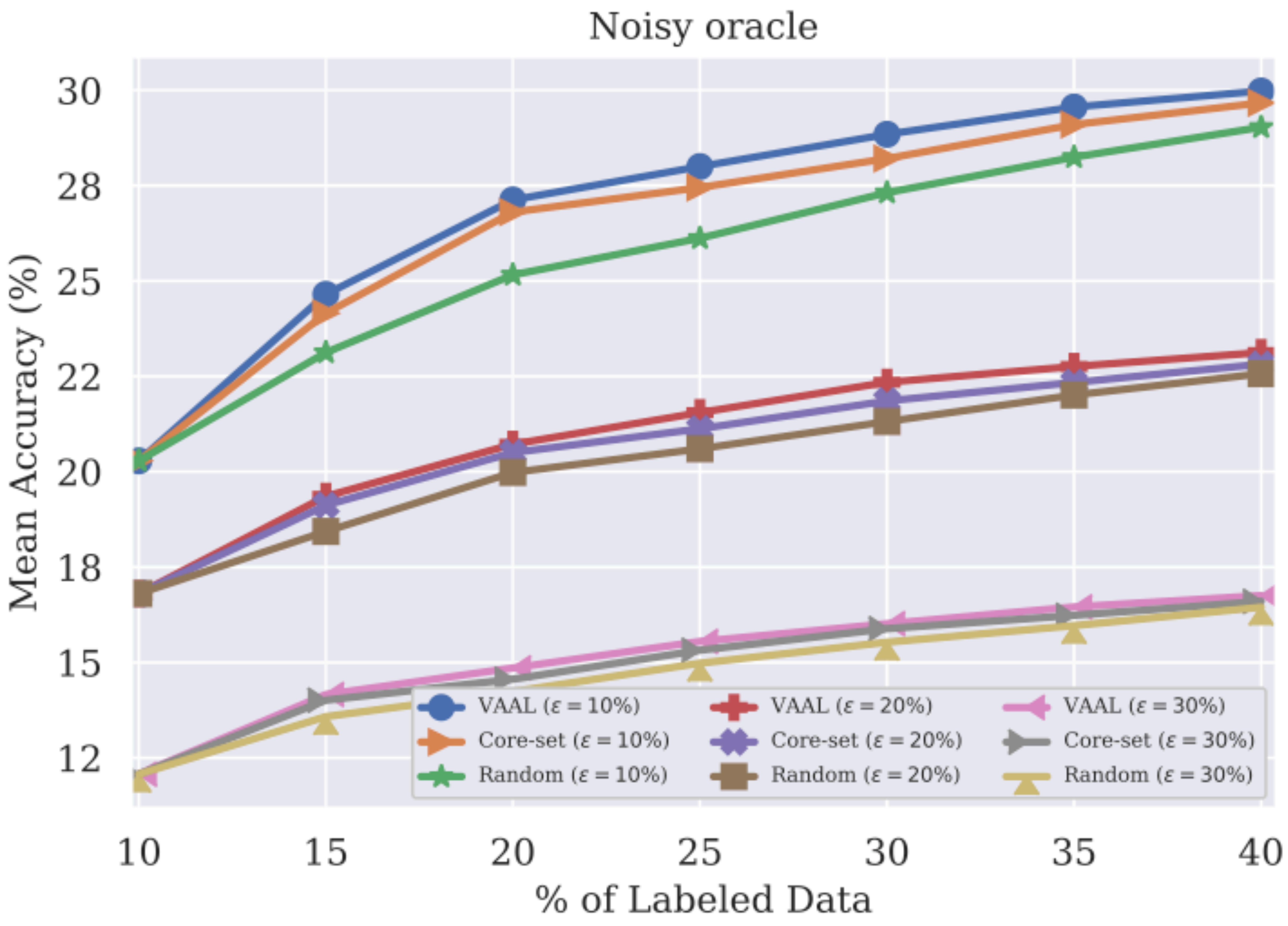
Recht et al, “Do ImageNet Classifiers Generalize to ImageNet?”, ICML 2019

Recall: noisy oracles

Recall our active learning assumptions:

- *Oracle is infallible:*
  the teacher/labeler does not make mistakes!

- *Pool belongs to task*:
  we will cover this in our lecture on "learning and the unknown"



Sinha et al, "Variational Adversarial Active Learning", ICCV 2019

Perspectives to address these challenges

More than known vs. unknown

1. **Known knowns:**

   Do you have an intuition what these 4 categories could represent?

2. **Known unknowns:**

3. **Unknown unknowns:**

4. **Unknown knowns:**

More than known vs. unknown

1. **Known knowns:**

   Examples belong to the distribution from training set was drawn. Assumption of an accurate & confident prediction.

2. **Known unknowns:**

3. **Unknown unknowns:**

4. **Unknown knowns:**

More than known vs. unknown

1. **Known knowns:**

   Examples belong to the distribution from training set was drawn. Assumption of an accurate & confident prediction.

2. **Known unknowns:**

   Unknown examples where models are not confident or uncertainty is high. Can be optionally “negatively” labelled examples used in training.

3. **Unknown unknowns:**

4. **Unknown knowns:**

More than known vs. unknown

1. **Known knowns:**

   Examples belong to the distribution from training set was drawn. Assumption of an accurate & confident prediction.

2. **Known unknowns:**

   Unknown examples where models are not confident or uncertainty is high. Can be optionally "negatively" labelled examples used in training.

3. **Unknown unknowns:**

   Unseen instances belonging to unexplored & unknown data distributions. Predictions generally overconfident & by definition false.

4. **Unknown knowns:**

More than known vs. unknown

1. **Known knowns (*or simply knowns*):**

   Examples belong to the distribution from training set was drawn. Assumption of an accurate & confident prediction.

2. **Known unknowns:**

   Unknown examples where models are not confident or uncertainty is high. Can be optionally "negatively" labelled examples used in training.

3. **Unknown unknowns:**

   Unseen instances belonging to unexplored & unknown data distributions. Predictions generally overconfident & by definition false.

4. **Unknown knowns:**

   Usually not considered: we know the concept but choose to treat it as unknown (willful ignorance?) or our ML system cannot represent the concept + structure altogether

Three types of approaches

What do you think: how can we solve our challenge?

Three types of approaches

Anomalies in predictions:
The *unsuspecting angle*, where out-of-distribution are hopefully separable through anomalous output values.

Incorporating prior knowledge:
The *intuitive idea* to include “background” or “non-example” data population explicitly.

Open Set recognition:
The more *formal approach* ensures that we only rely on predictions from our “covered space”; we create bounds.

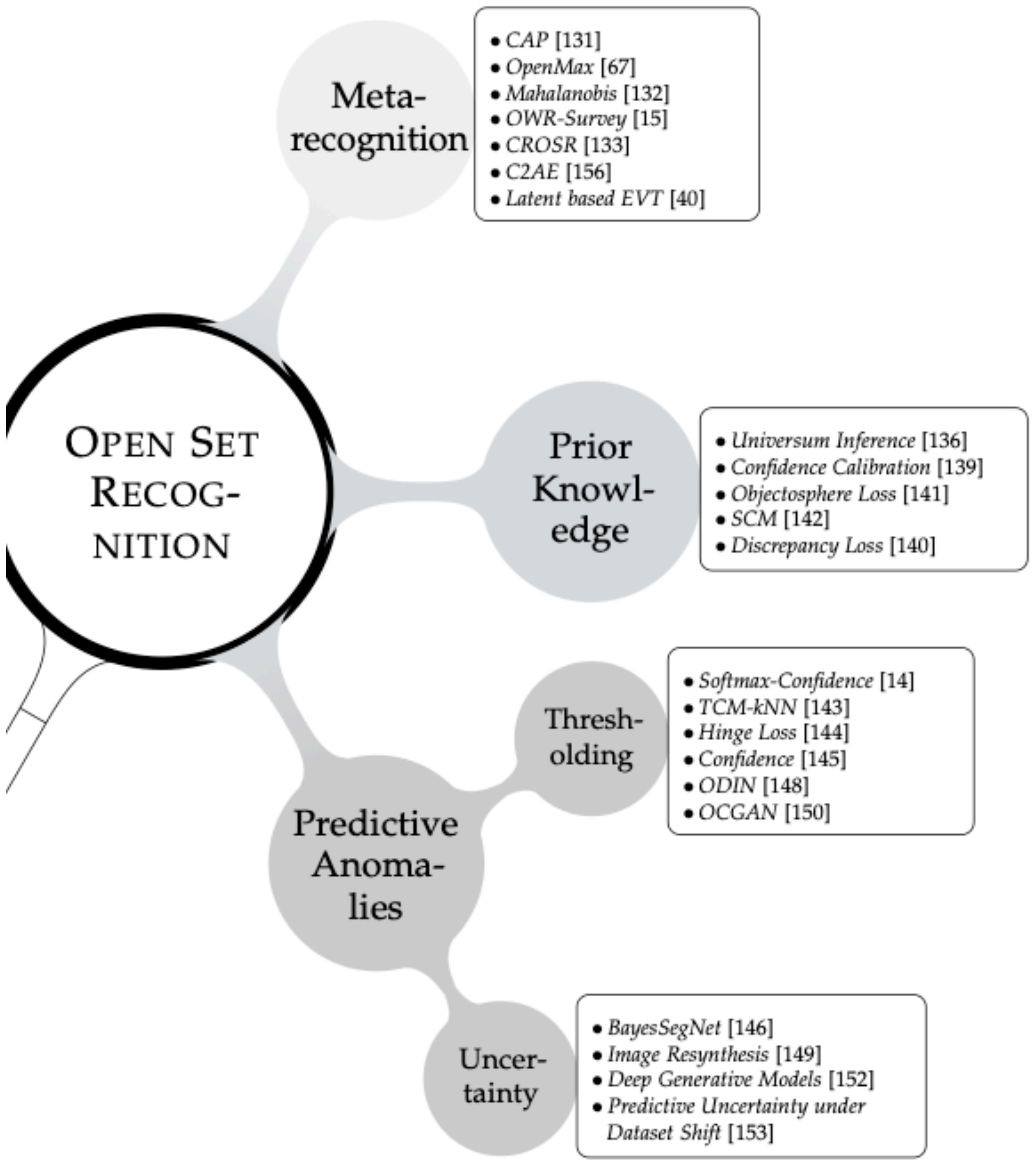
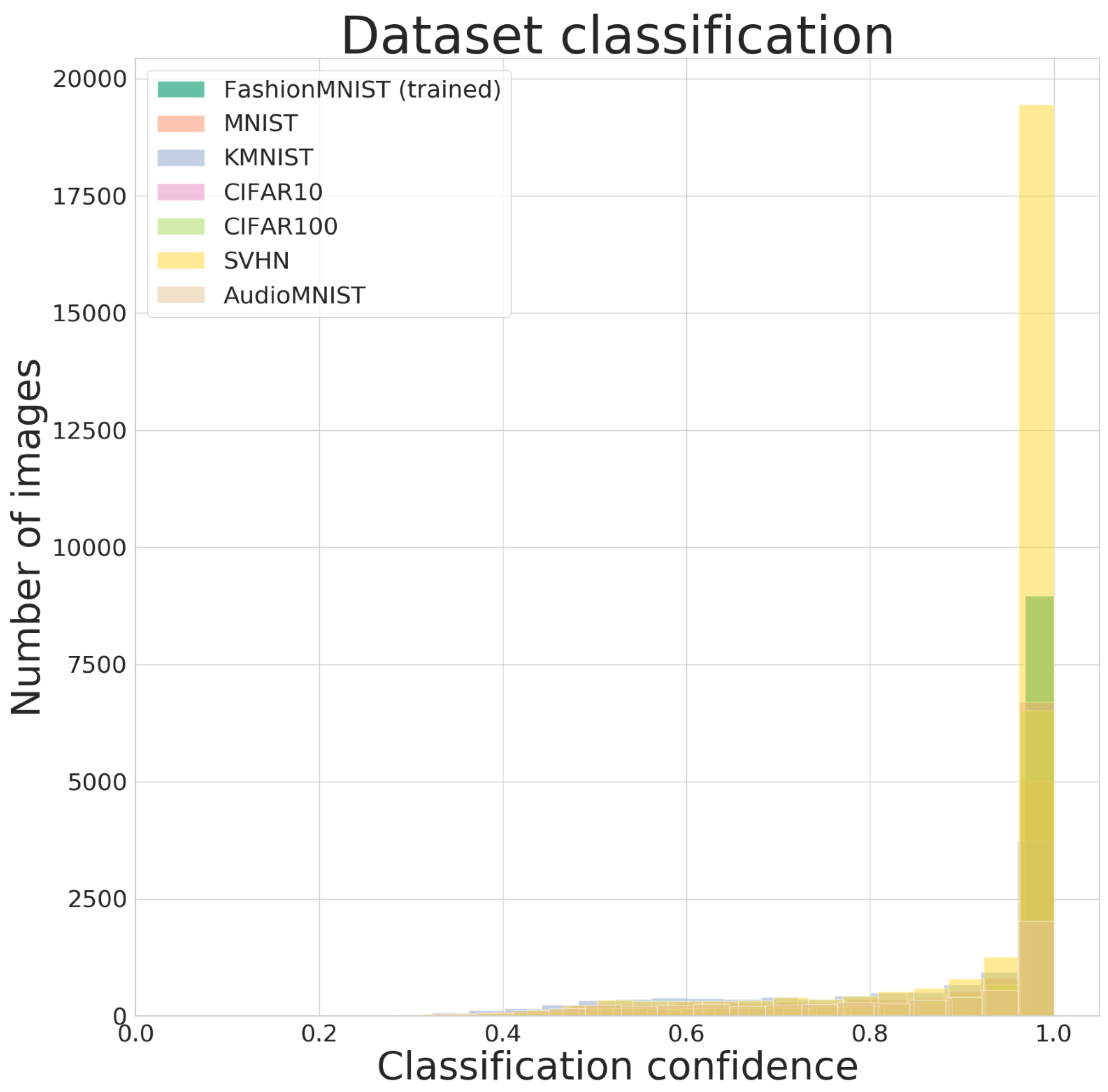


Figure from “A Wholistic View of Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning”, Mundt et al, Neural Networks, 2023

Predictive anomalies:
the unfortunate part of the story

Disclaimer: I'll use my many figures from our papers for convenience, without trying to imply that we discovered these phenomena

Recall lecture 1: overconfidence



Recall the quantitative example:

1. Train a neural network classifier on a dataset (here fashion items)
2. Log predictions for arbitrary other datasets
3. Observe that majority of misclassifications happen with large output “probability”

Overconfidence & uncertainty

Unfortunately uncertainty is not a necessarily a “fix”

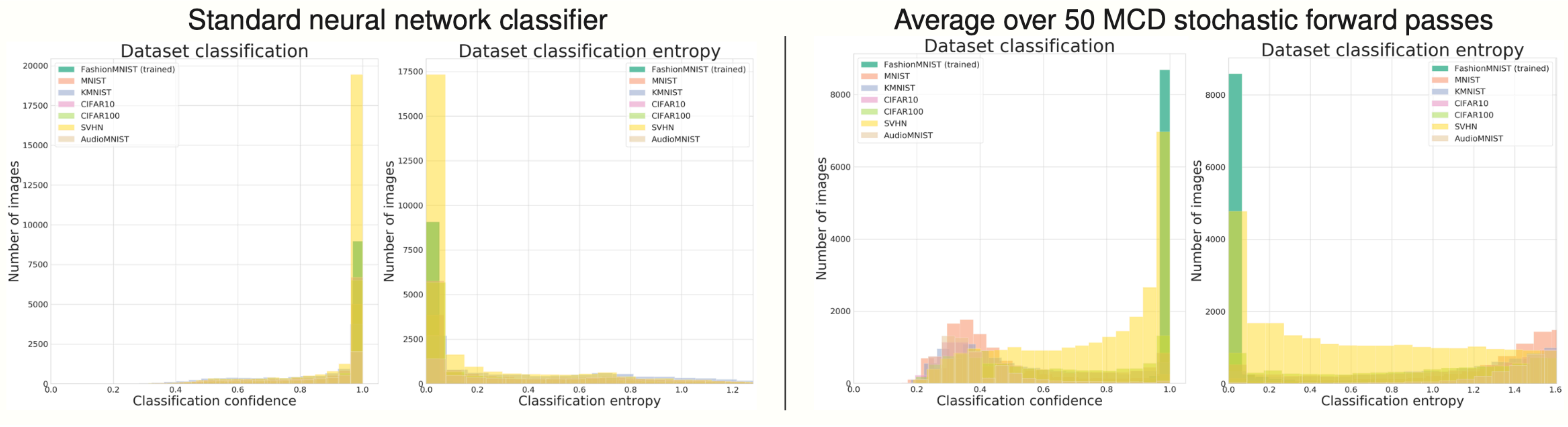
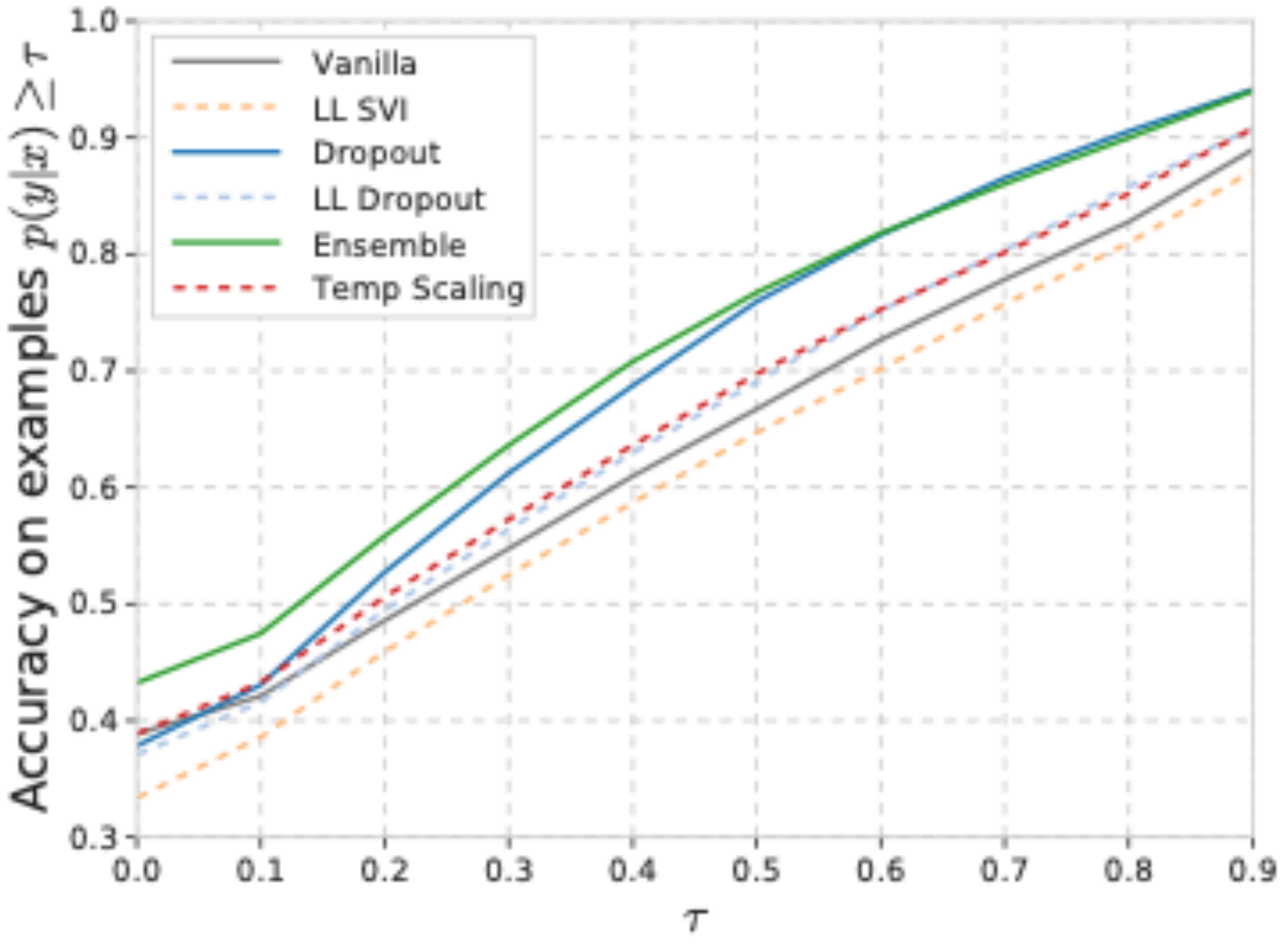


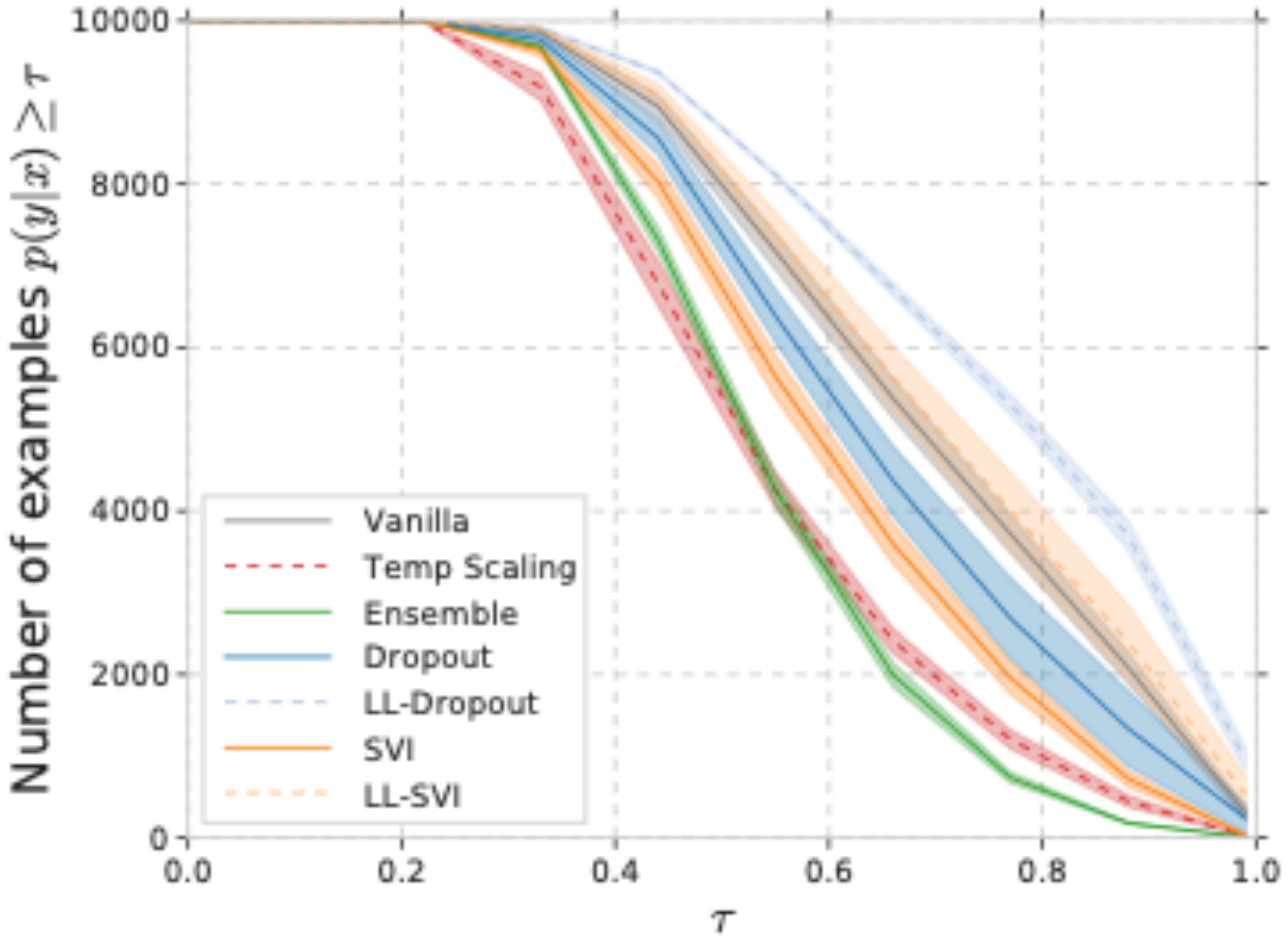
Figure from Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty, Does Out-of-Distribution Detection Require Generative Classifiers?”, ICCVW 2019

Overconfidence & uncertainty

Unfortunately uncertainty is not a necessarily a "fix"
& it get's even harder when we try to select a threshold



(d) ImageNet: Confidence vs Acc



(f) CIFAR: Confidence on OOD

Ovadia & Fertig et al, "Can you trust your model's uncertainty?" Evaluating predictive uncertainty under dataset shift", NeurIPS 2019

Overconfidence & gen. models

Unfortunately uncertainty is not a necessarily a “fix”
& it get’s even harder when we try to select a threshold

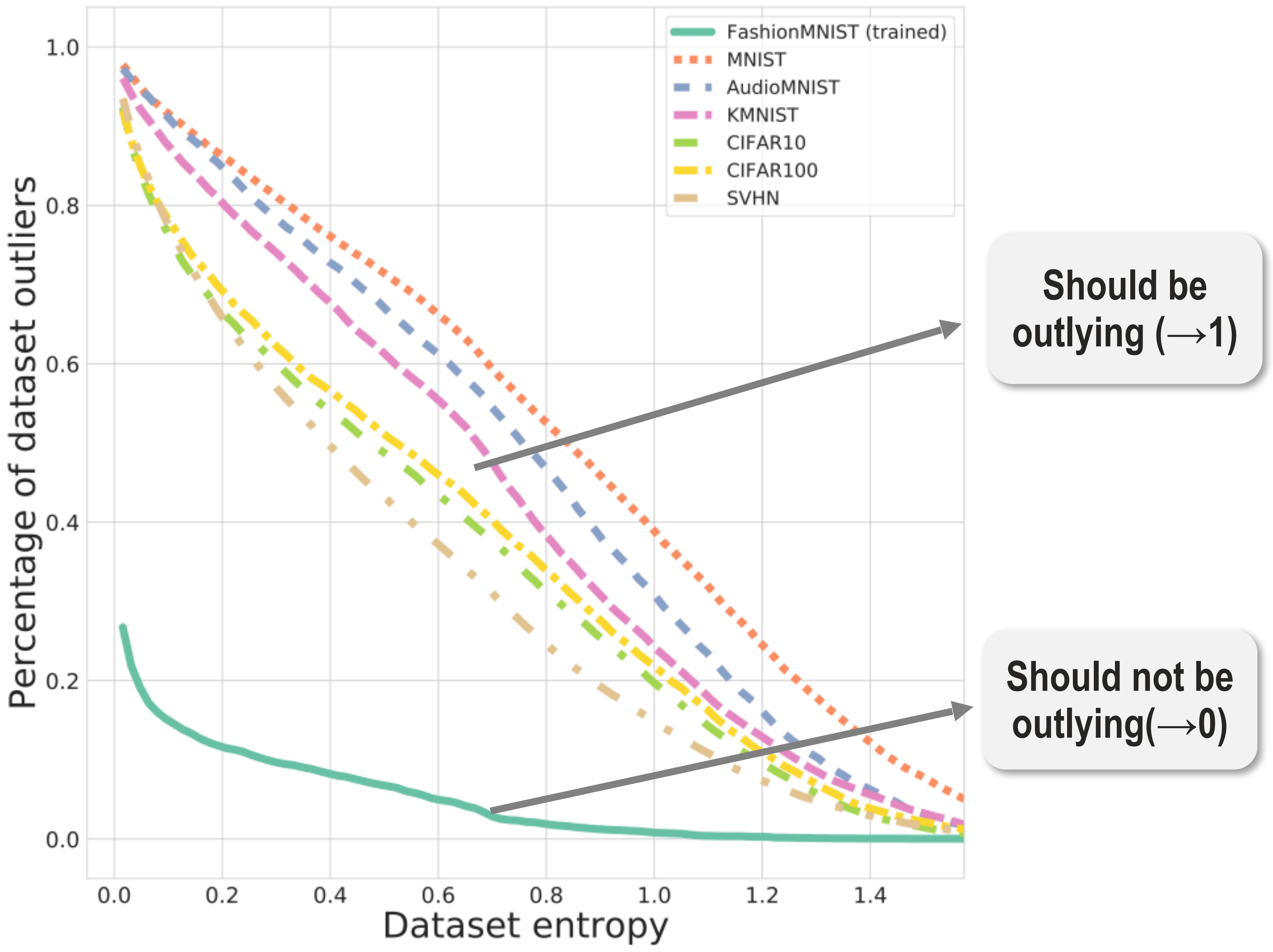


Figure from Mundt et al, “Unified Probabilistic Deep Continual Learning Through Open Set Recognition and Generative Replay”, Journal of Imaging, Volume 8, Issue 4, 2022

Overconfidence & uncertainty

Overconfidence is not exclusive to discriminative models

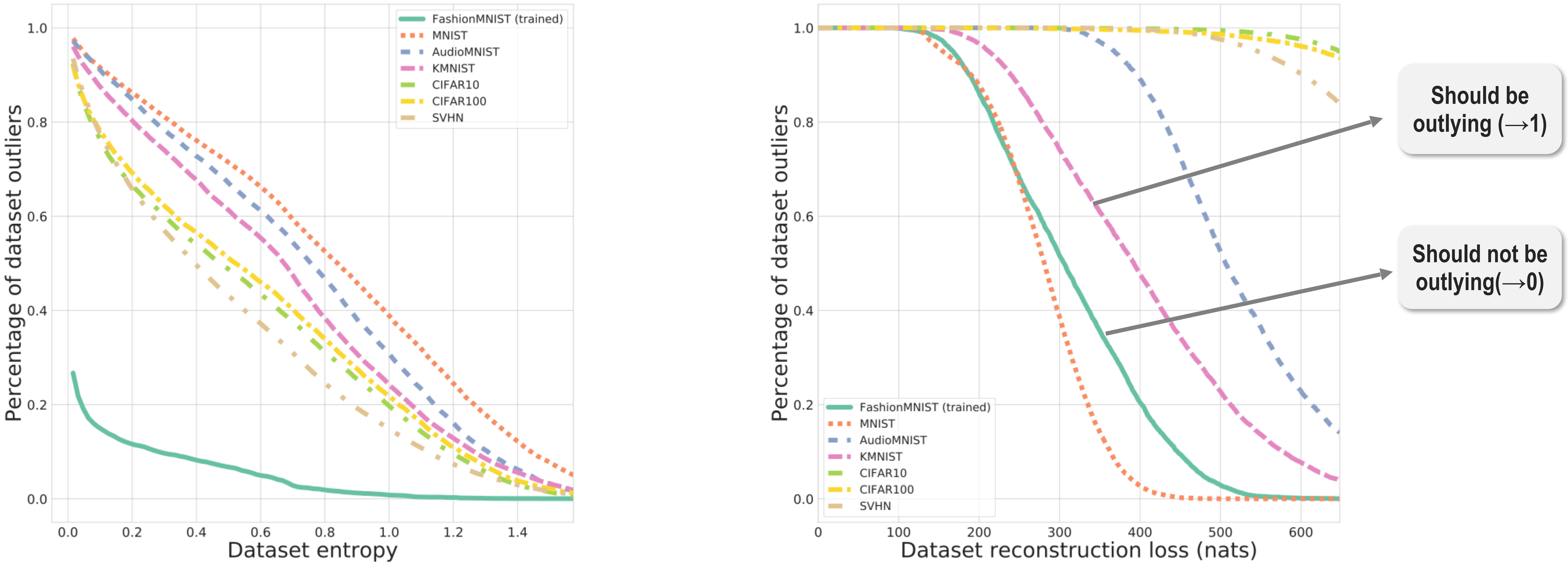
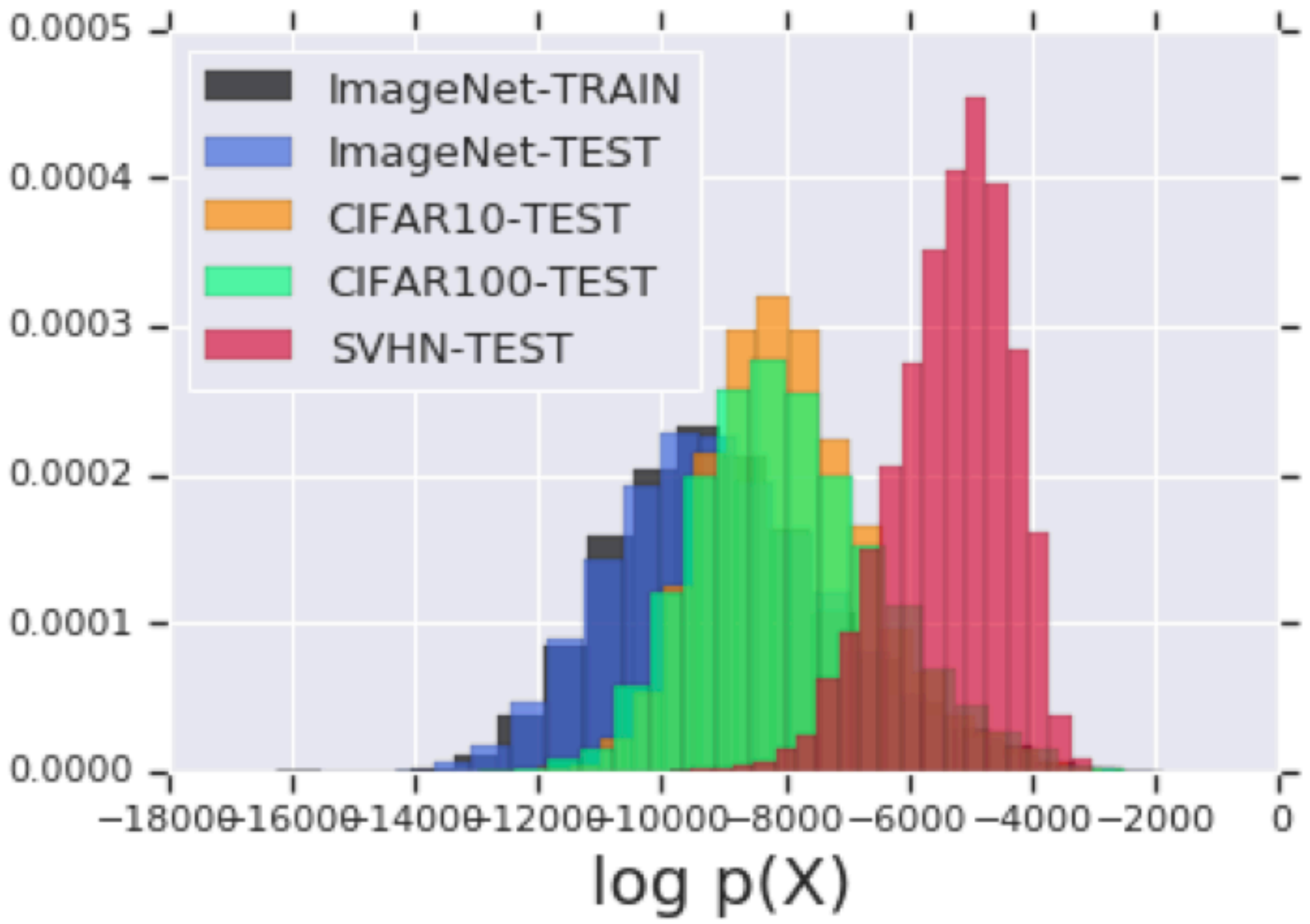


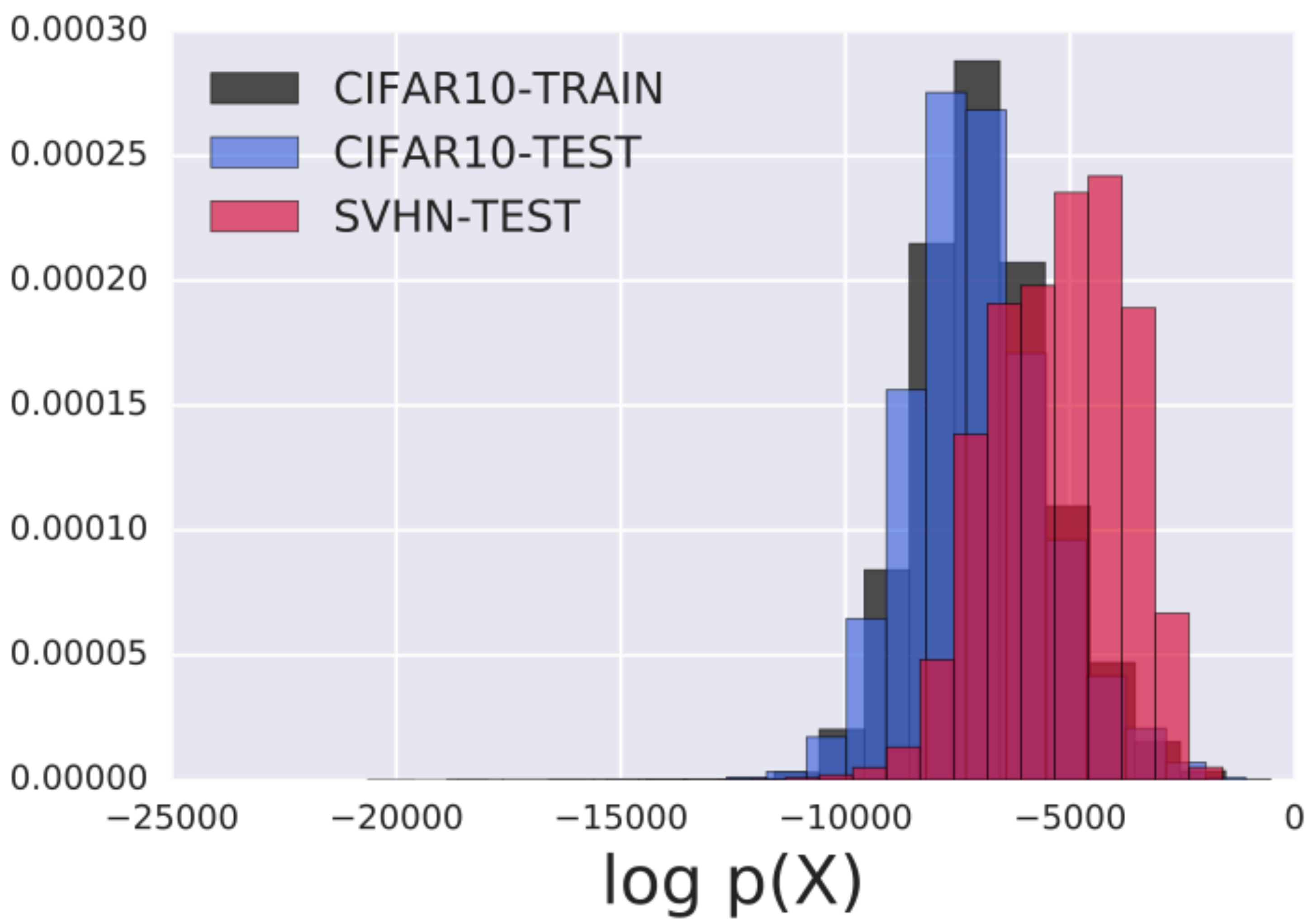
Figure from Mundt et al, “Unified Probabilistic Deep Continual Learning Through Open Set Recognition and Generative Replay”, Journal of Imaging, Volume 8, Issue 4, 2022

Overconfidence & gen. models

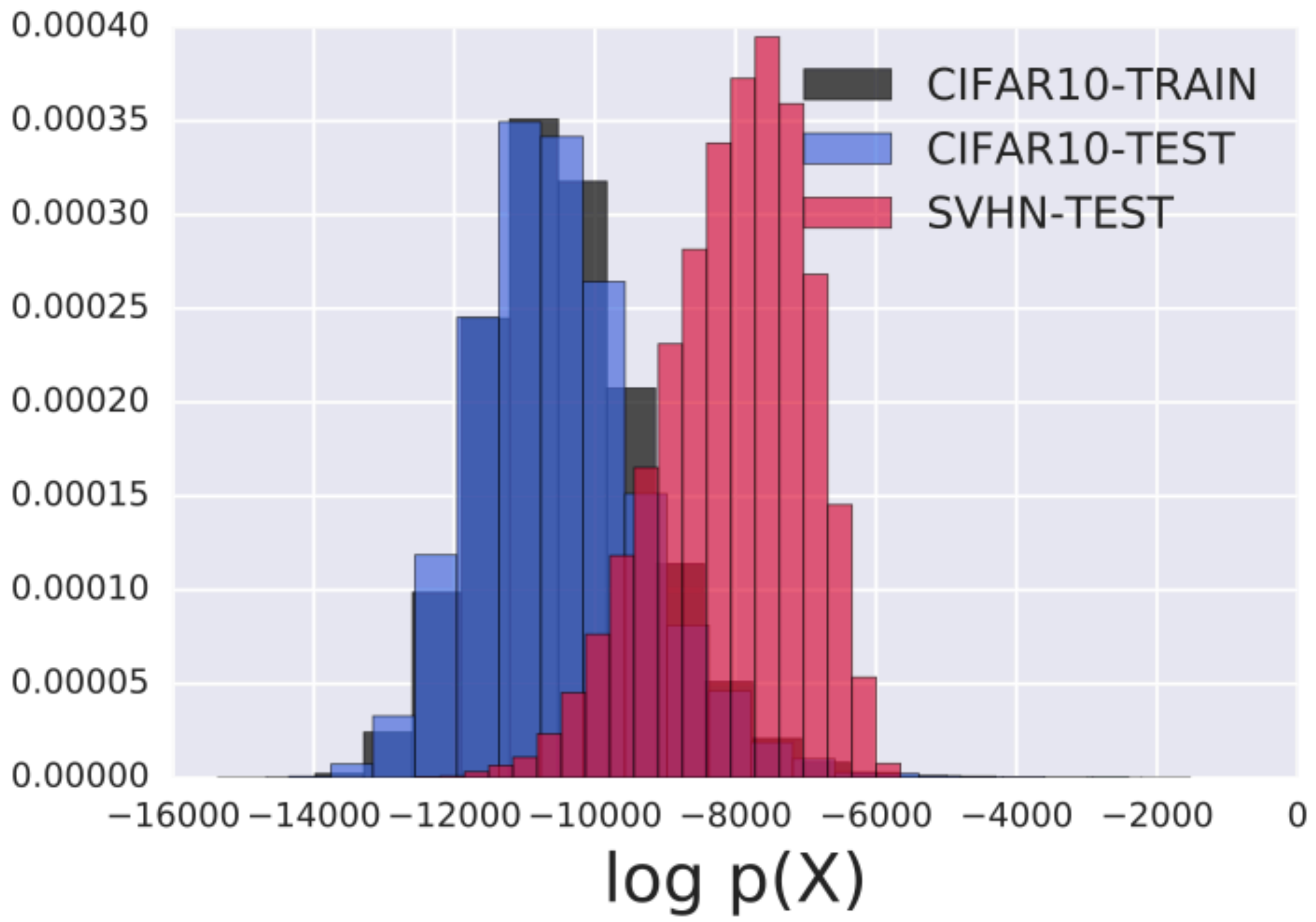
Overconfidence is not exclusive to discriminative models



Glow



PixelCNN



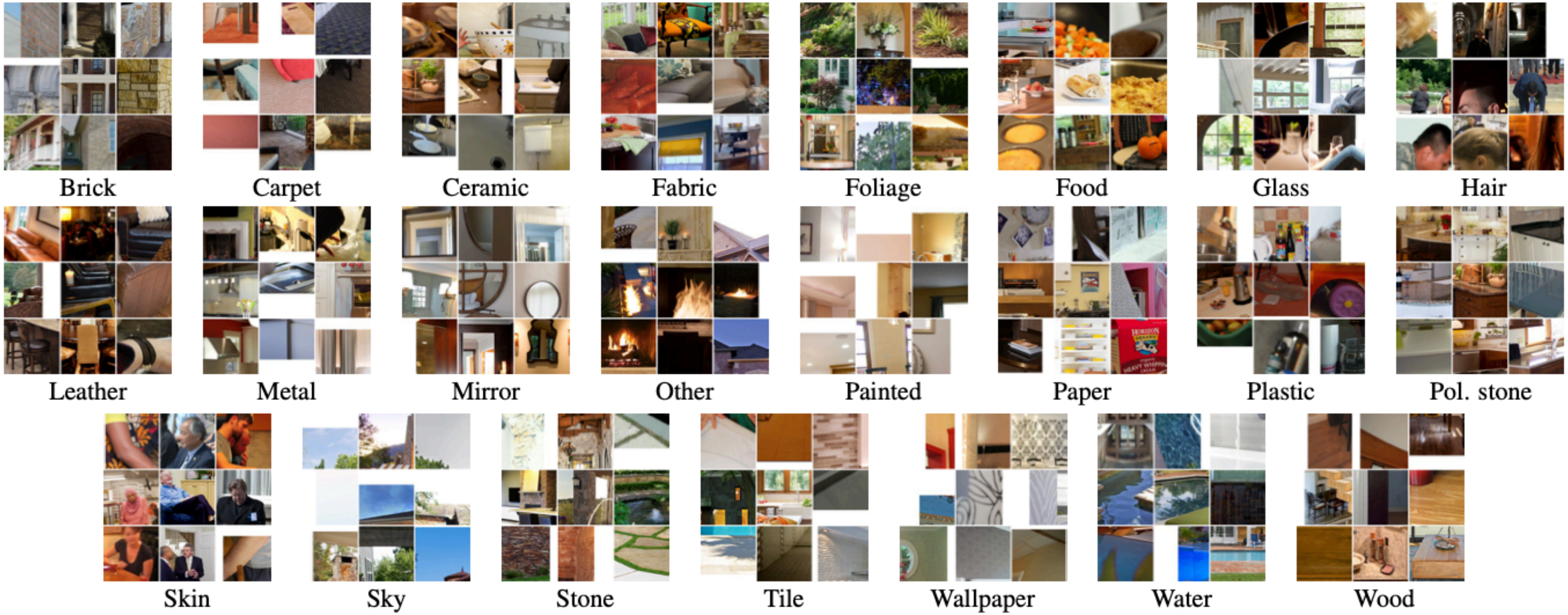
VAE

Nalisnick et al, “Do Deep Generative Models Know What They Don’t Know”, ICLR 2019

Including prior knowledge: an alternative?

The intuitive idea

Take a look at the below Materials in Context (MINC) dataset: what do you notice?



Bell & Upchurch et al, "Material Recognition in the Wild with the Materials in Context Database", CVPR 2015

The intuitive idea

An intuitive idea is to incorporate everything we know that does not belong to our task(s)

Bell & Upchurch et al, “Material Recognition in the Wild with the Materials in Context Database”, CVPR 2015

Inference with the universum

In essence: include **background class / “non-examples”** that aren’t of interest

Key questions:

- How to implement the loss: many many conceivable conceivable
  (Disclaimer: possibly *uncountable* amount of works)
- “what part of the universum is useful” (“Inference with the universum”, Weston et al, ICML 2006)
- "what are we expected to see during prediction later”? (Noise? Other concepts? Etc.)

Calibration: some examples

1. We could let our predictions (classifier) explicitly follow a uniform distribution for “out” data

(Kimin Lee et al, “Training confidence-calibrated classifiers for detecting out-of-distribution samples”, ICLR 2018)

$$\min_{\theta} \mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})} \left[-\log P_\theta \left(y = \widehat{y} | \widehat{\mathbf{x}} \right) \right] + \beta \mathbb{E}_{P_{\text{out}}(\mathbf{x})} \left[KL \left(\mathcal{U}\left(y\right) \parallel P_\theta \left(y | \mathbf{x}\right) \right) \right]$$

Calibration: some examples

1. We could let our predictions (classifier) explicitly follow a uniform distribution for “out” data

   (Kimin Lee et al, “Training confidence-calibrated classifiers for detecting out-of-distribution samples”, ICLR 2018)

$$\min_{\theta} \mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})}\left[-\log P_\theta\left(y=\widehat{y}|\widehat{\mathbf{x}}\right)\right] + \beta \mathbb{E}_{P_{\text{out}}(\mathbf{x})}\left[KL\left(\mathcal{U}\left(y\right) \parallel P_\theta\left(y|\mathbf{x}\right)\right)\right]$$

2. We could calibrate our outputs, e.g. by scaling a temperature parameter later

   (Liang et al, “Enhancing the reliability of out-of-distribution image detection in neural networks”, ICLR 2018)

$$S_i(\boldsymbol{x};T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^{N}\exp\left(f_j(\boldsymbol{x})/T\right)}$$

Calibration: some examples

1. We could let our predictions (classifier) explicitly follow a uniform distribution for "out" data

(Kimin Lee et al, "Training confidence-calibrated classifiers for detecting out-of-distribution samples", ICLR 2018)

$$\min_{\theta} \mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})} \left[-\log P_\theta \left(y = \widehat{y} | \widehat{\mathbf{x}} \right) \right] + \beta \mathbb{E}_{P_{\text{out}}(\mathbf{x})} \left[KL \left(\mathcal{U}(y) \parallel P_\theta \left(y | \mathbf{x} \right) \right) \right]$$

2. We could calibrate our outputs, e.g. by scaling a temperature parameter later

(Liang et al, "Enhancing the reliability of out-of-distribution image detection in neural networks", ICLR 2018)

$$S_i(\boldsymbol{x}; T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^{N} \exp\left(f_j(\boldsymbol{x})/T\right)}$$

3. And many other versions to modify our loss, e.g.:

(Dhamija et al, "Reducing network agnostophobia", NeurIPS 2018)

$$J_E(x) = \begin{cases} -\log S_c(x) & \text{if } x \in \mathcal{D}'_c \text{ is from class } c \\ -\frac{1}{C} \sum\limits_{c=1}^{C} \log S_c(x) & \text{if } x \in \mathcal{D}'_b \end{cases}$$

Background & Objectosphere

We could also think about encouraging features to be zero for OOD data

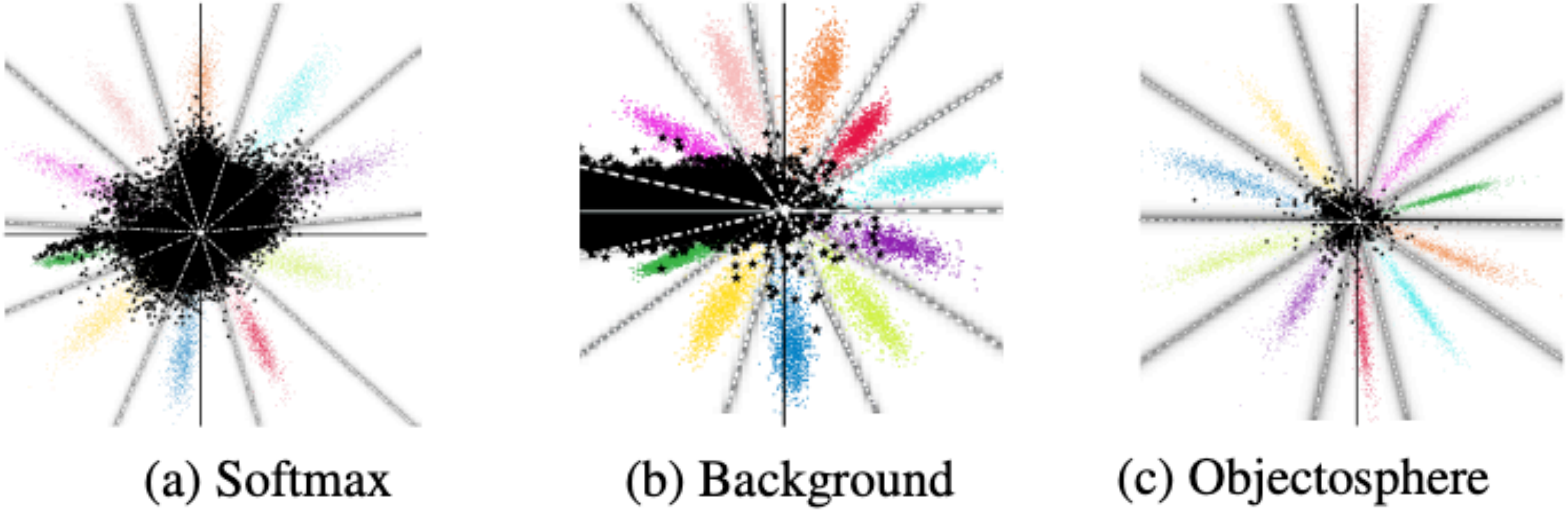


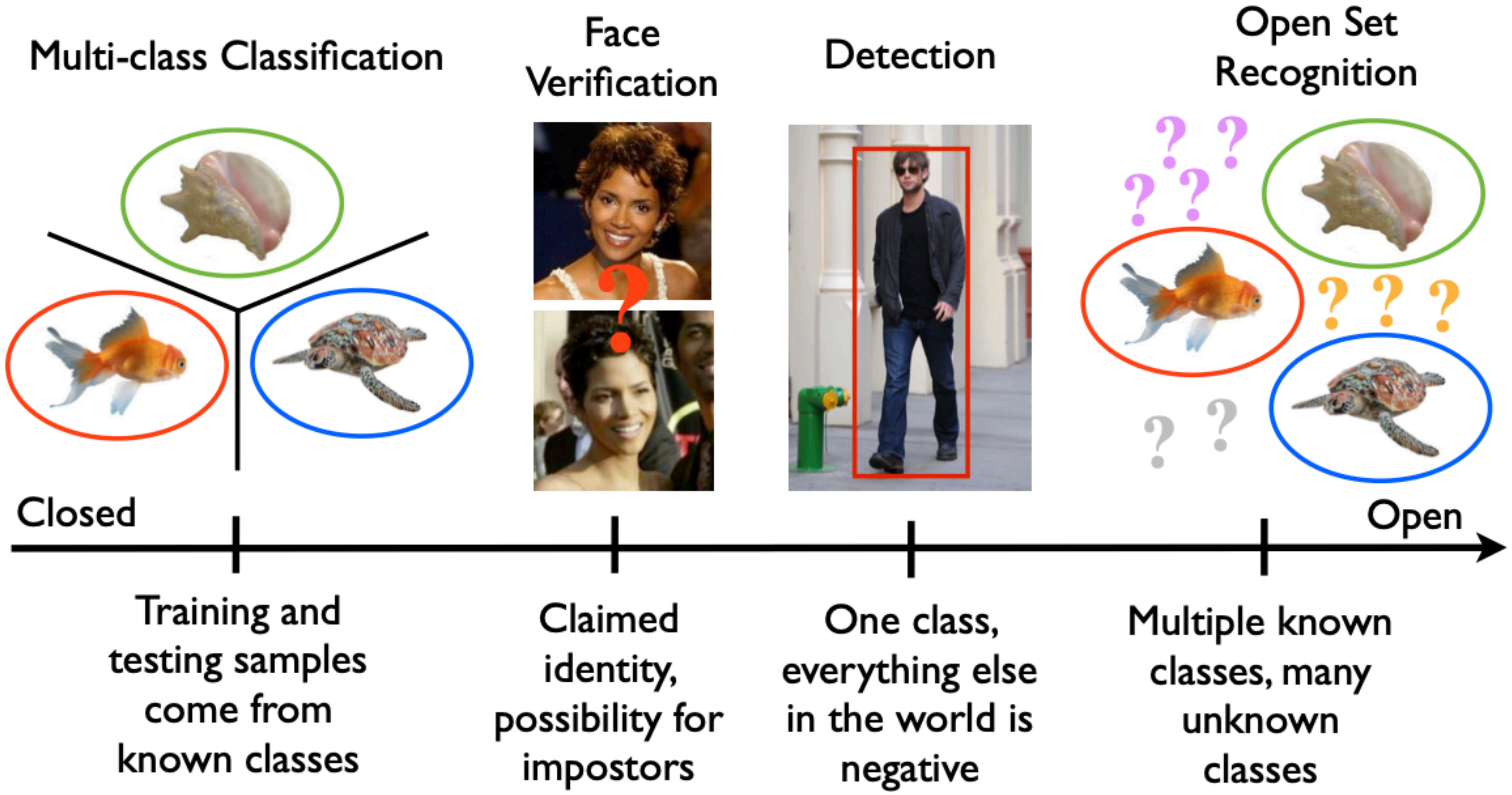
Figure 1: LENET++ RESPONSES TO KNOWNS AND UNKNOWNS. *The network in (a) was only trained to classify the 10 MNIST classes ($\mathcal{D}'_c$) using softmax, while the networks in (b) and (c) added NIST letters [15] as known unknowns ($\mathcal{D}'_b$) trained with softmax or our novel Objectosphere loss.*

Dhamija et al, "Reducing Network Agnostophobia", NeurIPS 2018

What do you think are the up & downsides so far?

Closed -open world assumption

We may need a different approach: as the world grows more "open" we move from known unknowns to unknown unknowns. Our two perspectives only handle the former

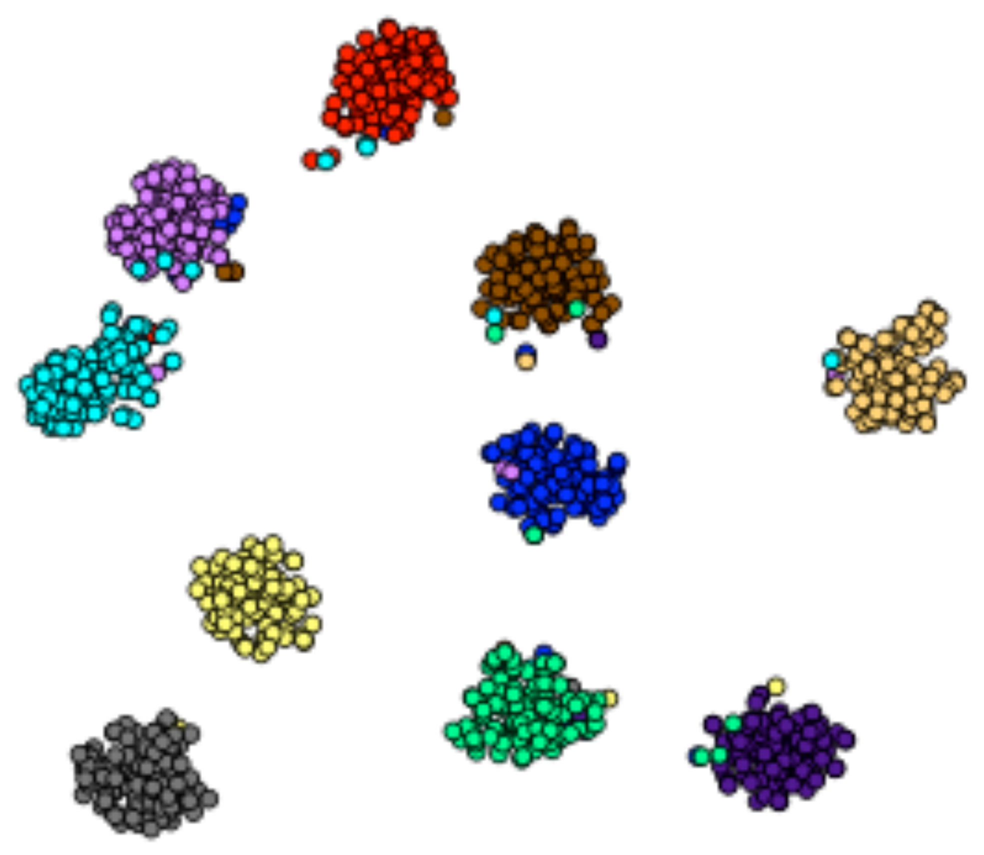


Scheirer et al, "Towards Open Set Recognition", TPAMI 2012

Open set recognition & explicit bounds

Intuition behind open space

Intuitively: we could take into account distances from the known data points



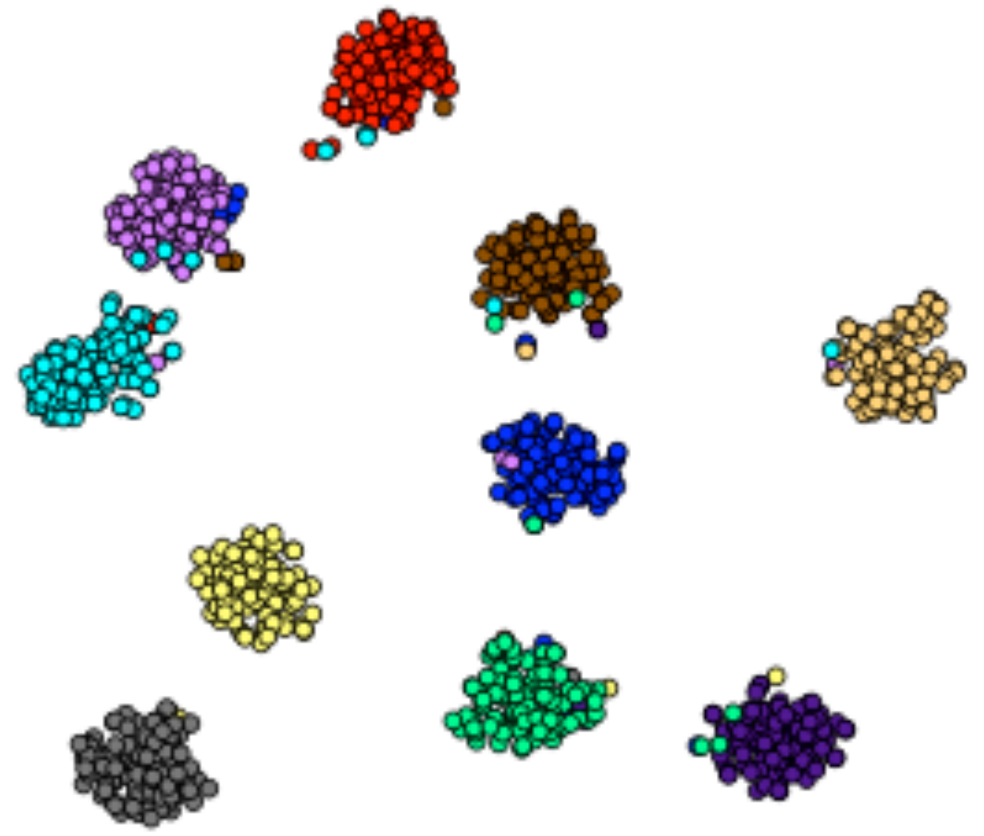
Intuition behind open space

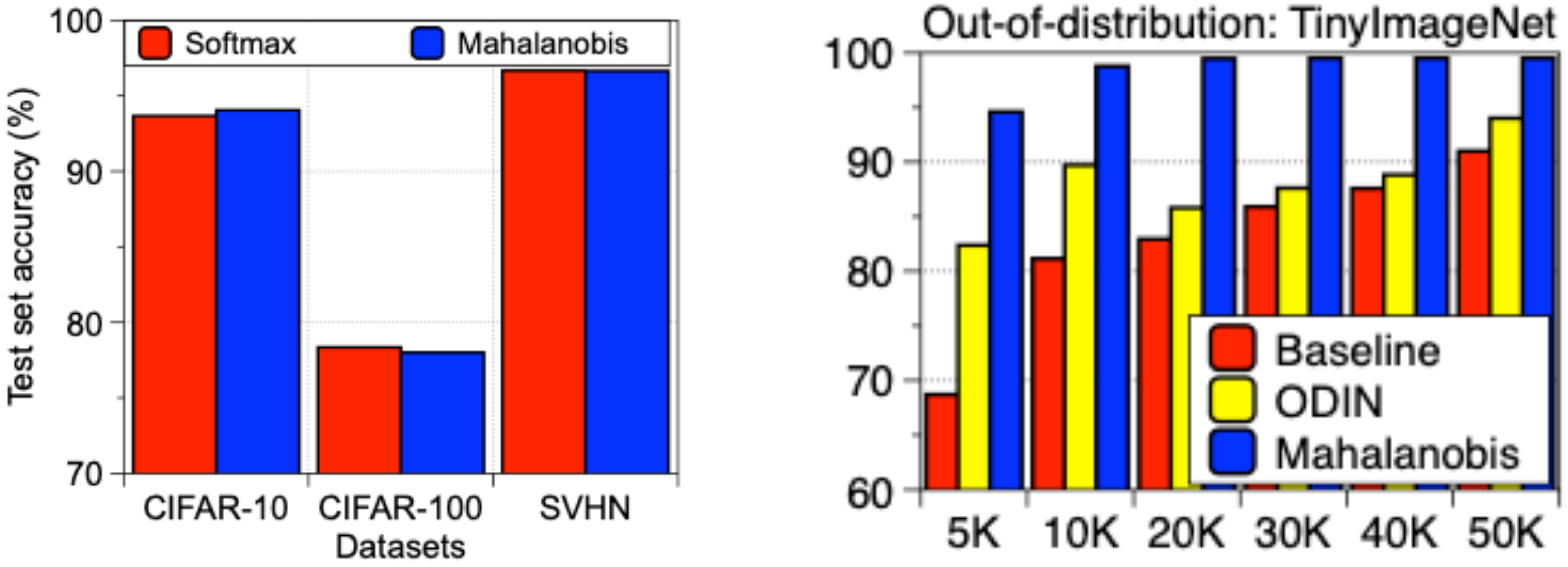
Intuitively: we could take into account distances from the known data points

Example 1 : we could make assumptions like every class being Normal distributed & then calculate distances to our existing data points, e.g. Mahalanobis distance

$$\widehat{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} f(\mathbf{x}_i), \quad \widehat{\mathbf{\Sigma}} = \frac{1}{N} \sum_c \sum_{i:y_i=c} \left(f(\mathbf{x}_i) - \widehat{\mu}_c\right)\left(f(\mathbf{x}_i) - \widehat{\mu}_c\right)^\top$$

$$M(\mathbf{x}) = \max_c \; -\left(f(\mathbf{x}) - \widehat{\mu}_c\right)^\top \widehat{\mathbf{\Sigma}}^{-1} \left(f(\mathbf{x}) - \widehat{\mu}_c\right)$$



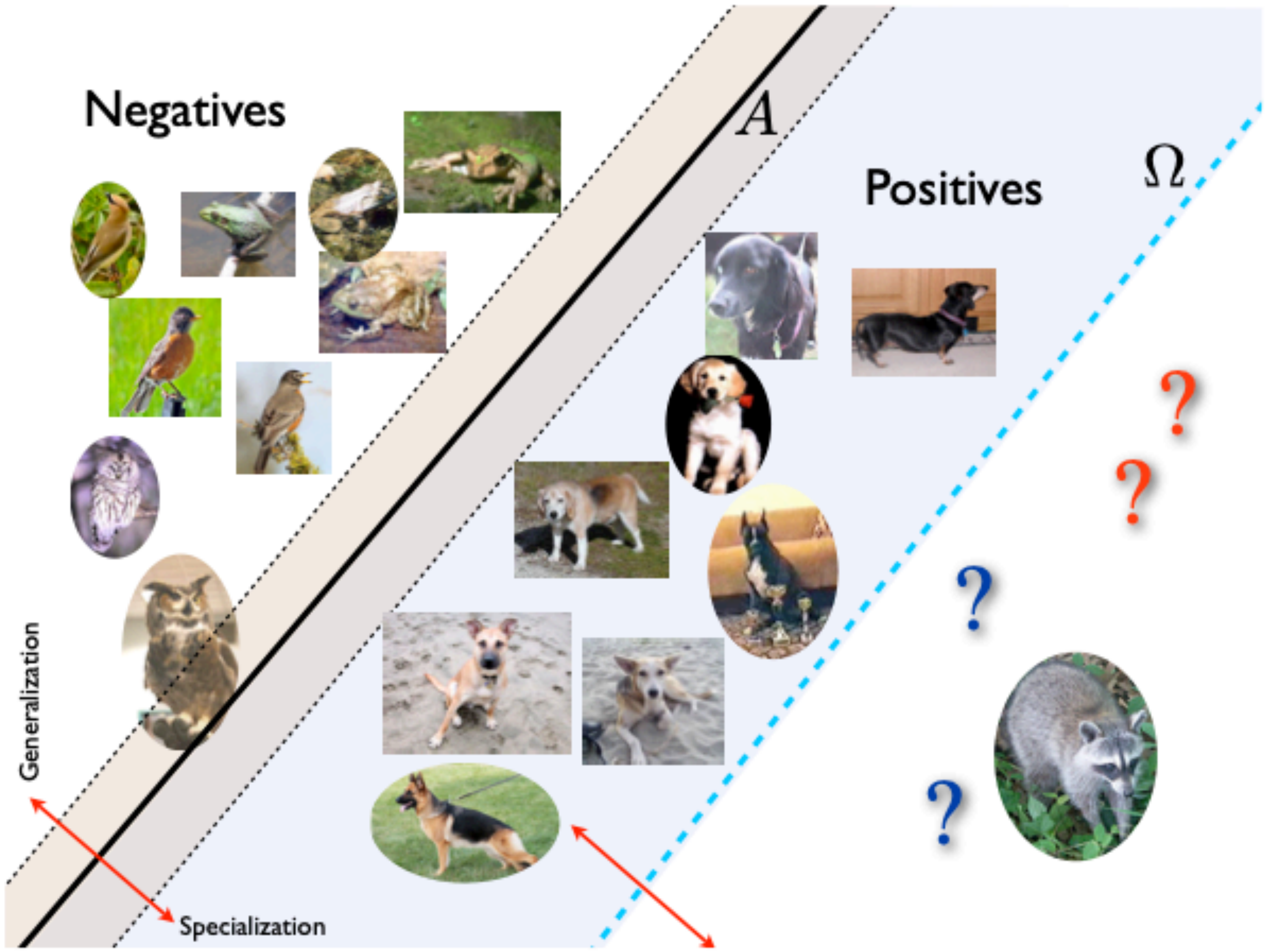


(E.g. Kimin Lee et al, “A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks”, NeurIPS 2018)

Intuition behind open space

Intuitively: we could take into account distances from the known data points

Example 2: we could fit another parallel plane in an SVM, for a reject option, based on the support set with large distances



Scheirer et al, "Towards Open Set Recognition", TPAMI 2012

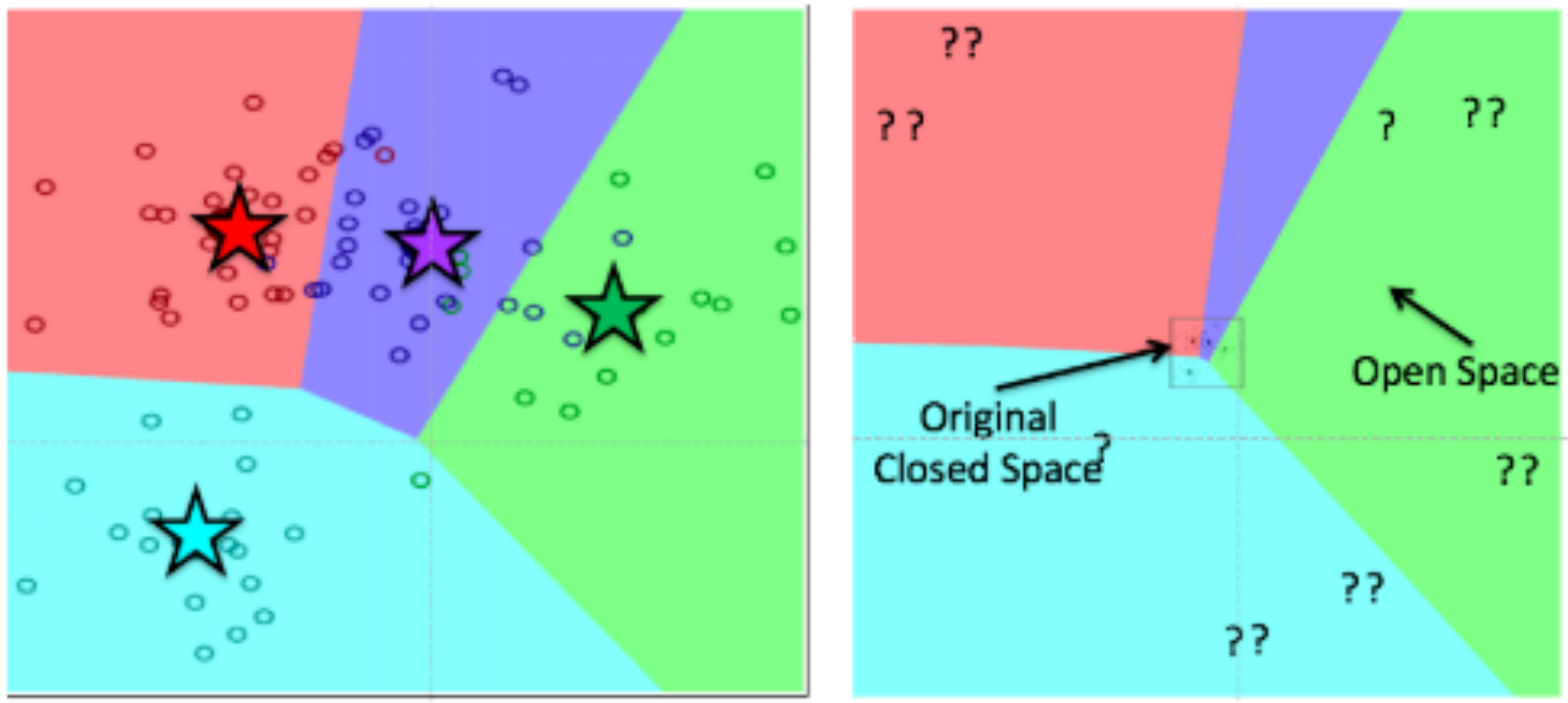
Formalizing open space/sets

Intuitively: open space is what we have not covered with known data

Formally: (see e.g. “Learning and the Unknown”, Boult et al, AAAI 2019)

For a recognition function function f over space $\mathscr{X}$ & a union of balls with radius r that includes all known training examples:

$$\mathscr{O} = \mathscr{X} - \cup_{i \in N} B_r(x_i)$$



“Learning and the Unknown”, Boult et al, AAAI 2019

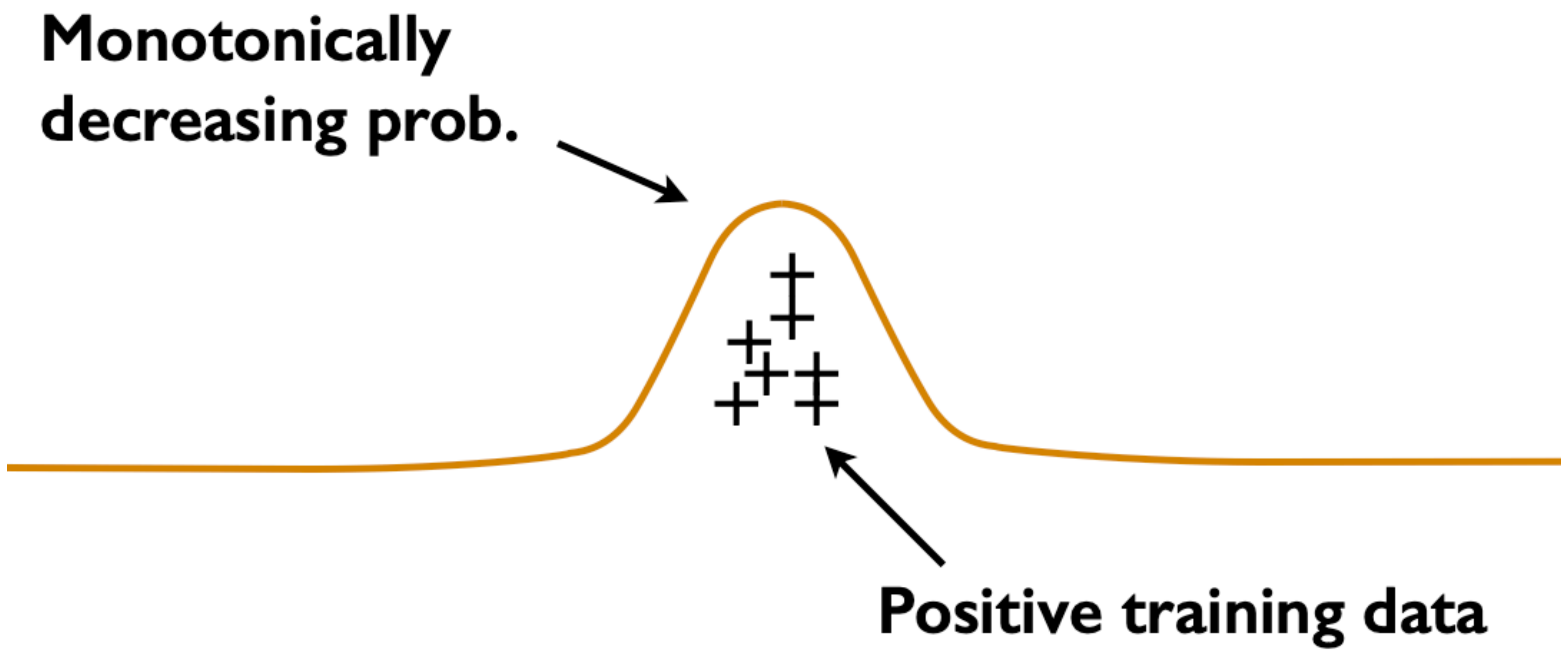
Formalizing open space/sets

For a recognition function function f over space $\mathcal{X}$ & a union of balls with radius r that includes all known training examples:

$$\mathcal{O} = \mathcal{X} - \cup_{i \in N} B_r(x_i)$$

Can now define open space risk as a relative measure of open space to the full space, but see the survey for the full math

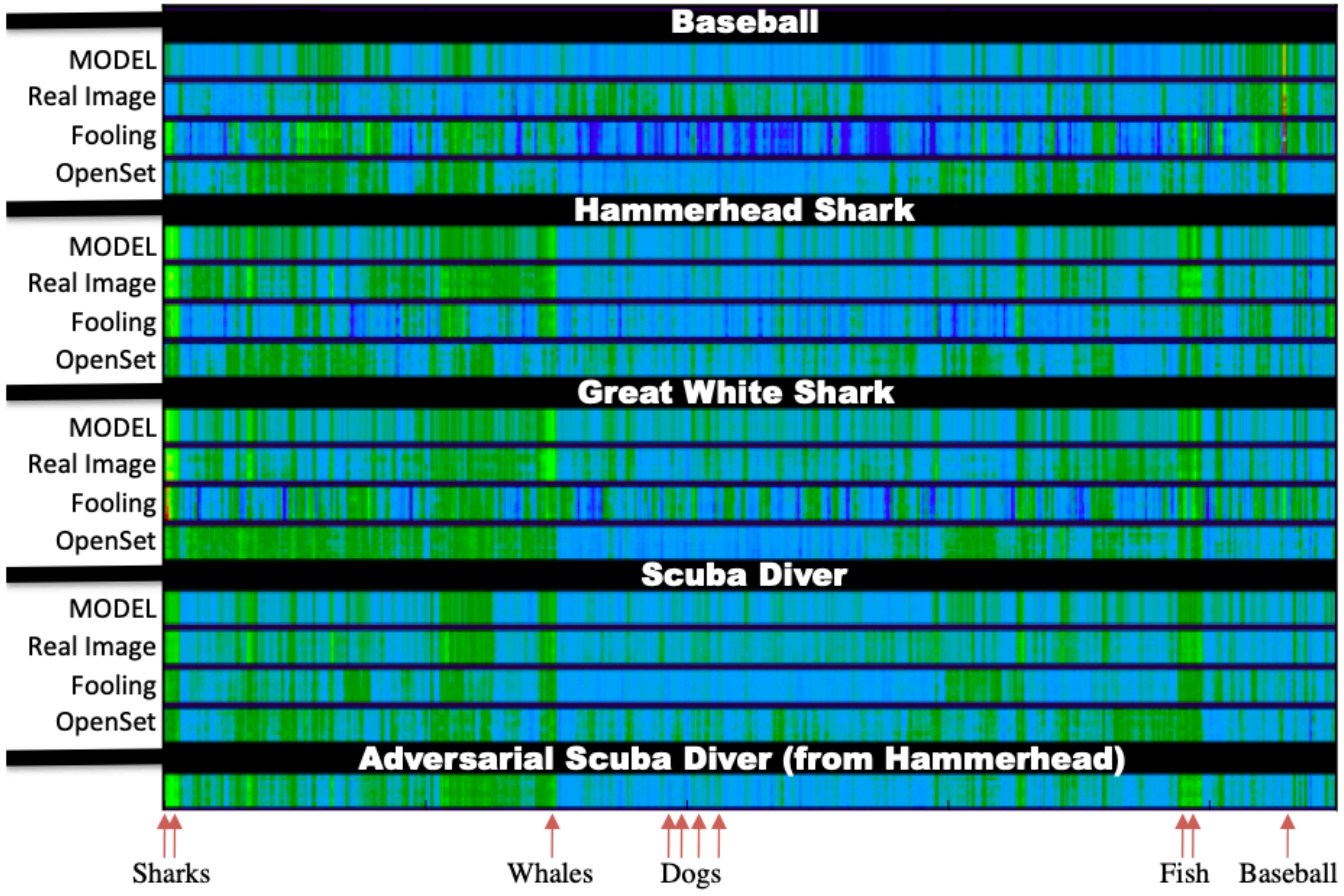
For now: the aim would be to decay the probability away from supporting evidence



Scheirer et al, "Probability Models for Open Set Recognition", TPAMI 2014

Bounds with extreme values

In other words, we could fit a distance based model (following the radius idea), e.g. here based on the mean activations of training data in a deep net



Bendale & Boult et al, “Towards Open Set Deep Networks”, CVPR 2016

Bounds with extreme values

In other words, we could fit a distance based model (following the radius idea), e.g. here based on the mean activations of training data in a deep net

Algorithm 1 EVT Meta-Recognition Calibration for Open Set Deep Networks, with per class Weibull fit to η largest distance to mean activation vector. Returns libMR models ρ_j which includes parameters τ_i for shifting the data as well as the Weibull shape and scale parameters:κ_i, λ_i.

Require: FitHigh function from libMR
Require: Activation levels in the penultimate network layer $\mathbf{v}(\mathbf{x}) = v_1(x) \ldots v_N(x)$
Require: For each class j let $S_{i,j} = v_j(x_{i,j})$ for each correctly classified training example $x_{i,j}$.

for $j = 1 \ldots N$ **do**
  Compute mean AV, $\mu_j = mean_i(S_{i,j})$
  EVT Fit $\rho_j = (\tau_j, \kappa_j, \lambda_j) = \text{FitHigh}(\|\hat{S}_j - \mu_j\|, \eta)$
end for
Return means μ_j and libMR models ρ_j

Bendale & Boult et al, “Towards Open Set Deep Networks”, CVPR 2016

Bounds with extreme values

In other words, we could fit a distance based model (following the radius idea), e.g. here based on the mean activations of training data in a deep net

But which distribution should we choose?

- We are mainly interested in the extreme distances, as we want to make a decision of when to reject
- Extreme value theory may provide an answer for us

Algorithm 1 EVT Meta-Recognition Calibration for Open Set Deep Networks, with per class Weibull fit to η largest distance to mean activation vector. Returns libMR models ρ_j which includes parameters τ_i for shifting the data as well as the Weibull shape and scale parameters: κ_i, λ_i.

Require: FitHigh function from libMR
Require: Activation levels in the penultimate network layer $\mathbf{v}(\mathbf{x}) = v_1(x) \dots v_N(x)$
Require: For each class j let $S_{i,j} = v_j(x_{i,j})$ for each correctly classified training example $x_{i,j}$.
1: **for** $j = 1 \dots N$ **do**
2:     **Compute mean AV,** $\mu_j = mean_i(S_{i,j})$
3:     **EVT Fit** $\rho_j = (\tau_j, \kappa_j, \lambda_j) = \text{FitHigh}(\|\hat{S}_j - \mu_j\|, \eta)$
4: **end for**
5: **Return** means μ_j and libMR models ρ_j

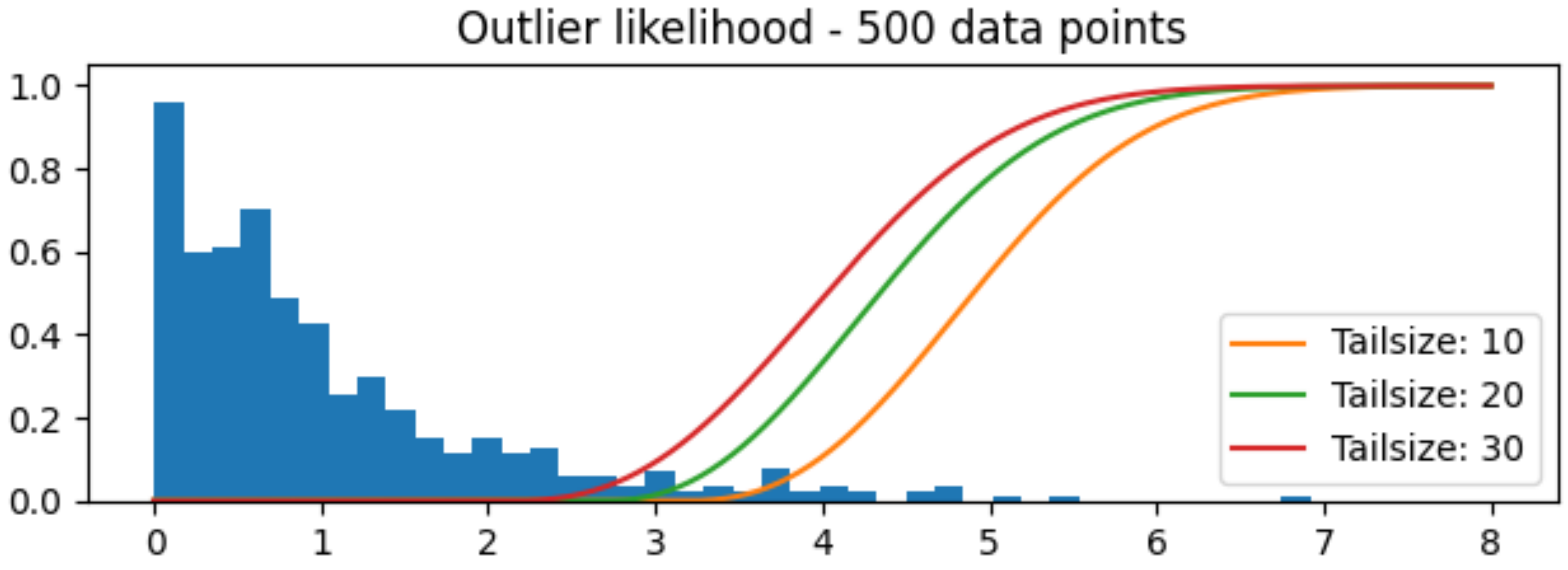
Bendale & Boult et al, "Towards Open Set Deep Networks", CVPR 2016

Bounds with extreme values

Extreme value theory is interested in the probability of events that are more extreme than any previously observed

$$f(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Regardless of the overall distribution, if the data is bounded, EVT tells us that sampling the tail/the extrema away from the median of our distribution results in an EVT distribution: Weibull, Gumbel or Fréchet



Bounds with extreme values

We can use the cumulative distribution function (CDF) to either reject right away, because we exceed our extremely observed distances, or use the value to modify our prediction score (Referred to as OpenMax here)

Algorithm 2 OpenMax probability estimation with rejection of unknown or uncertain inputs.

Require: Activation vector for $\mathbf{v}(\mathbf{x}) = v_1(x), \ldots, v_N(x)$
Require: **means** μ_j and libMR models $\rho_j = (\tau_i, \lambda_i, \kappa_i)$
Require: α, the numer of "top" classes to revise

Let $s(i) = \text{argsort}(v_j(x))$; Let $\omega_j = 1$
for $i = 1, \ldots, \alpha$ **do**

$$\omega_{s(i)}(x) = 1 - \frac{\alpha - i}{\alpha} e^{-\left(\frac{\|x - \tau_{s(i)}\|}{\lambda_{s(i)}}\right)^{\kappa_{s(i)}}}$$

end for
Revise activation vector $\hat{v}(x) = \mathbf{v}(\mathbf{x}) \circ \omega(\mathbf{x})$
Define $\hat{v}_0(x) = \sum_i v_i(x)(1 - \omega_i(x))$.

$$\hat{P}(y = j|\mathbf{x}) = \frac{e^{\hat{\mathbf{v}}_\mathbf{j}(\mathbf{x})}}{\sum_{i=0}^{N} e^{\hat{\mathbf{v}}_\mathbf{i}(\mathbf{x})}}$$

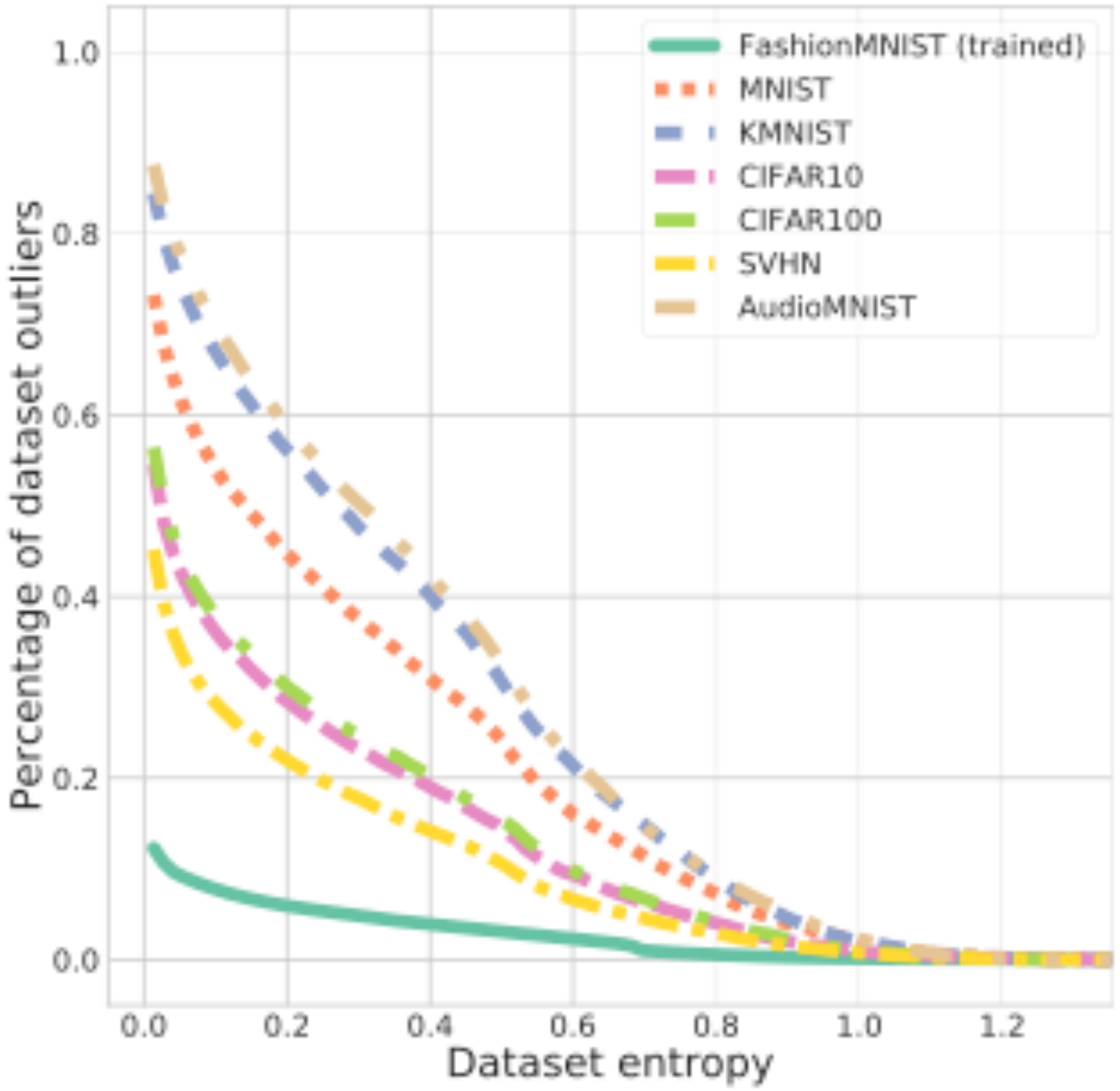
Let $y^* = \text{argmax}_j P(y = j|\mathbf{x})$
Reject input if $y^* == 0$ or $P(y = y^*|\mathbf{x}) < \epsilon$

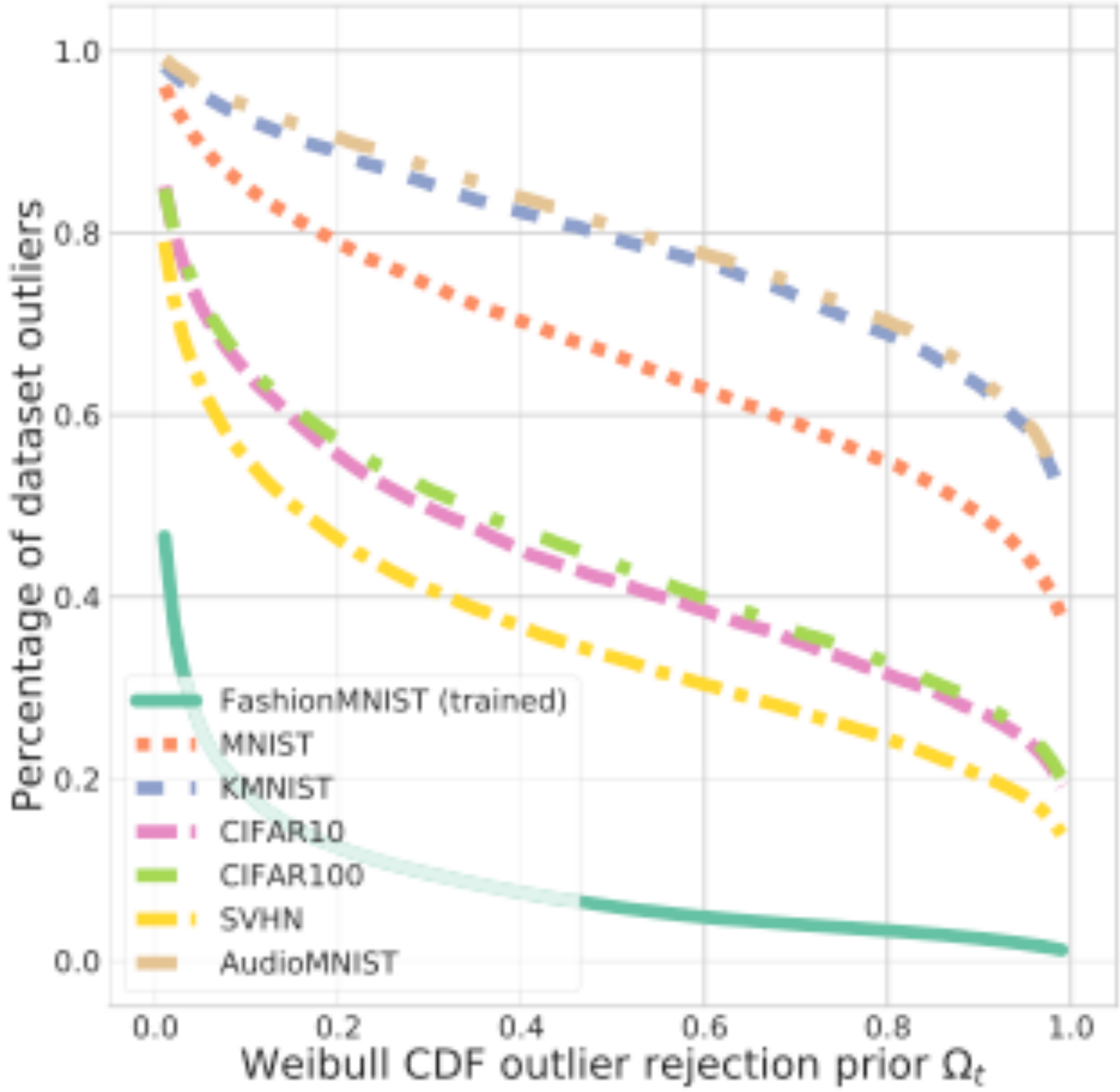
Bendale & Boult et al, "Towards Open Set Deep Networks", CVPR 2016

OpenMax in a generative variant

OpenMax seem to improve a lot!

But why is there still so much room for improvement?

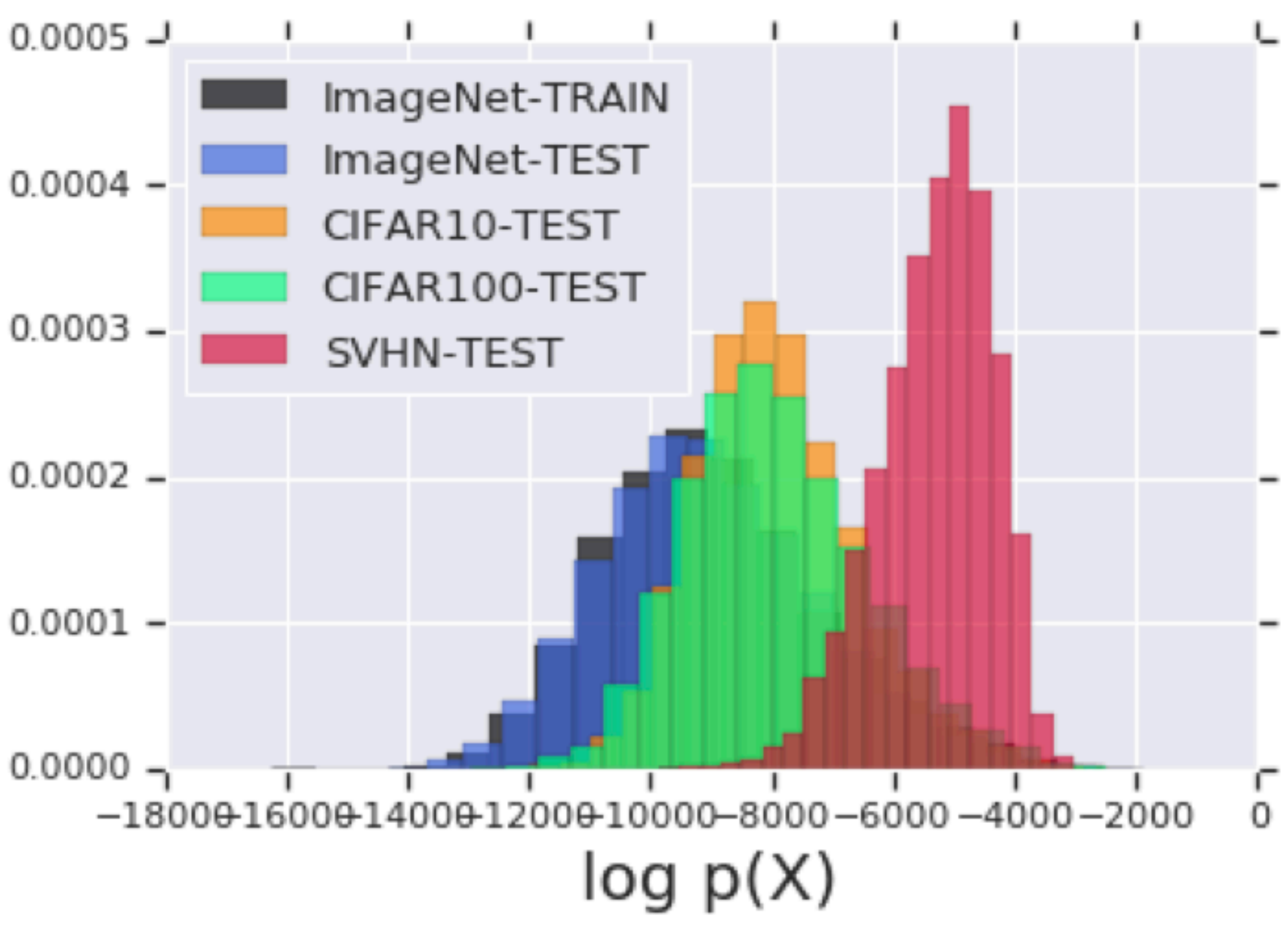




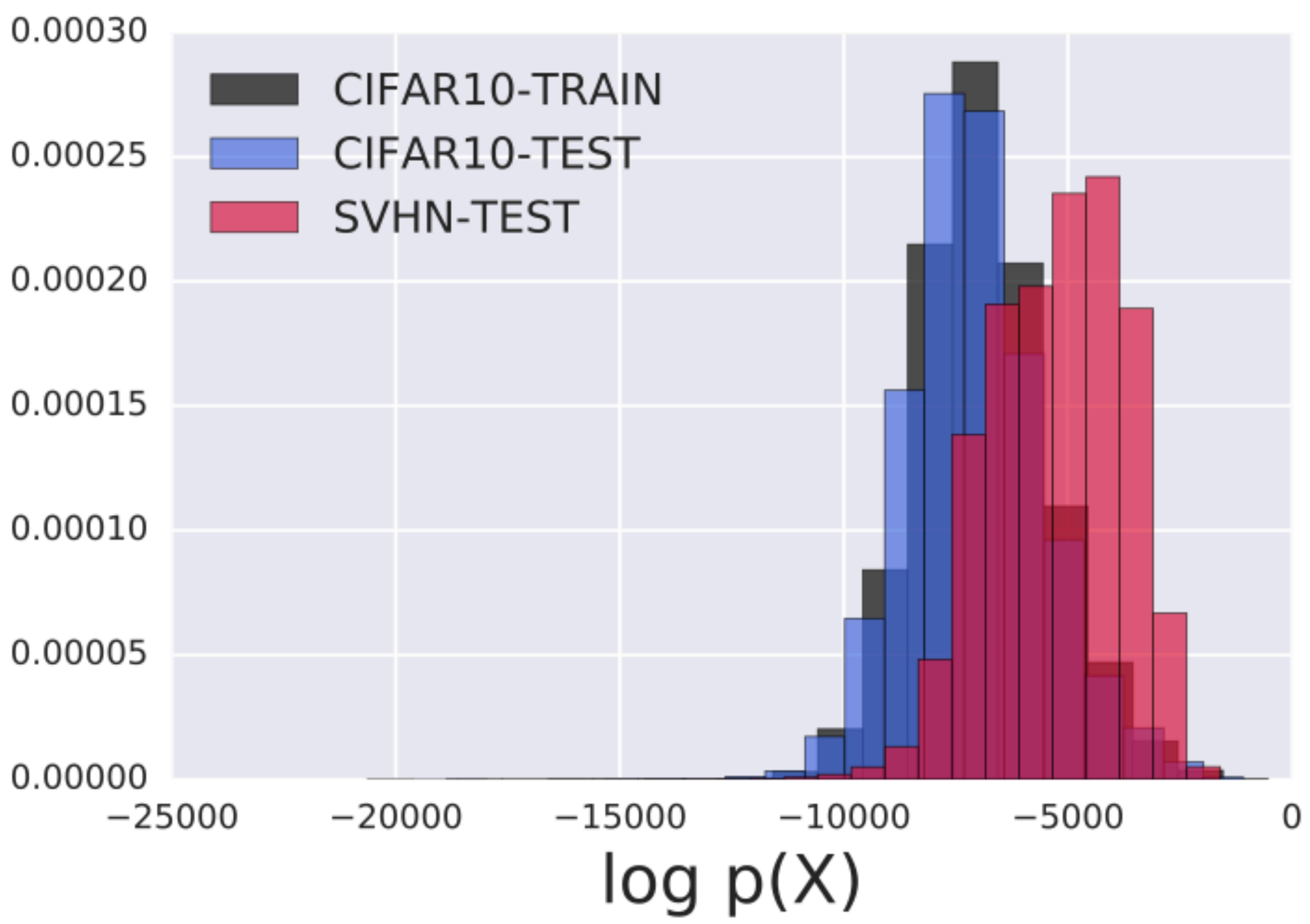
Do we need generative models on top?

Overconfidence & gen. models

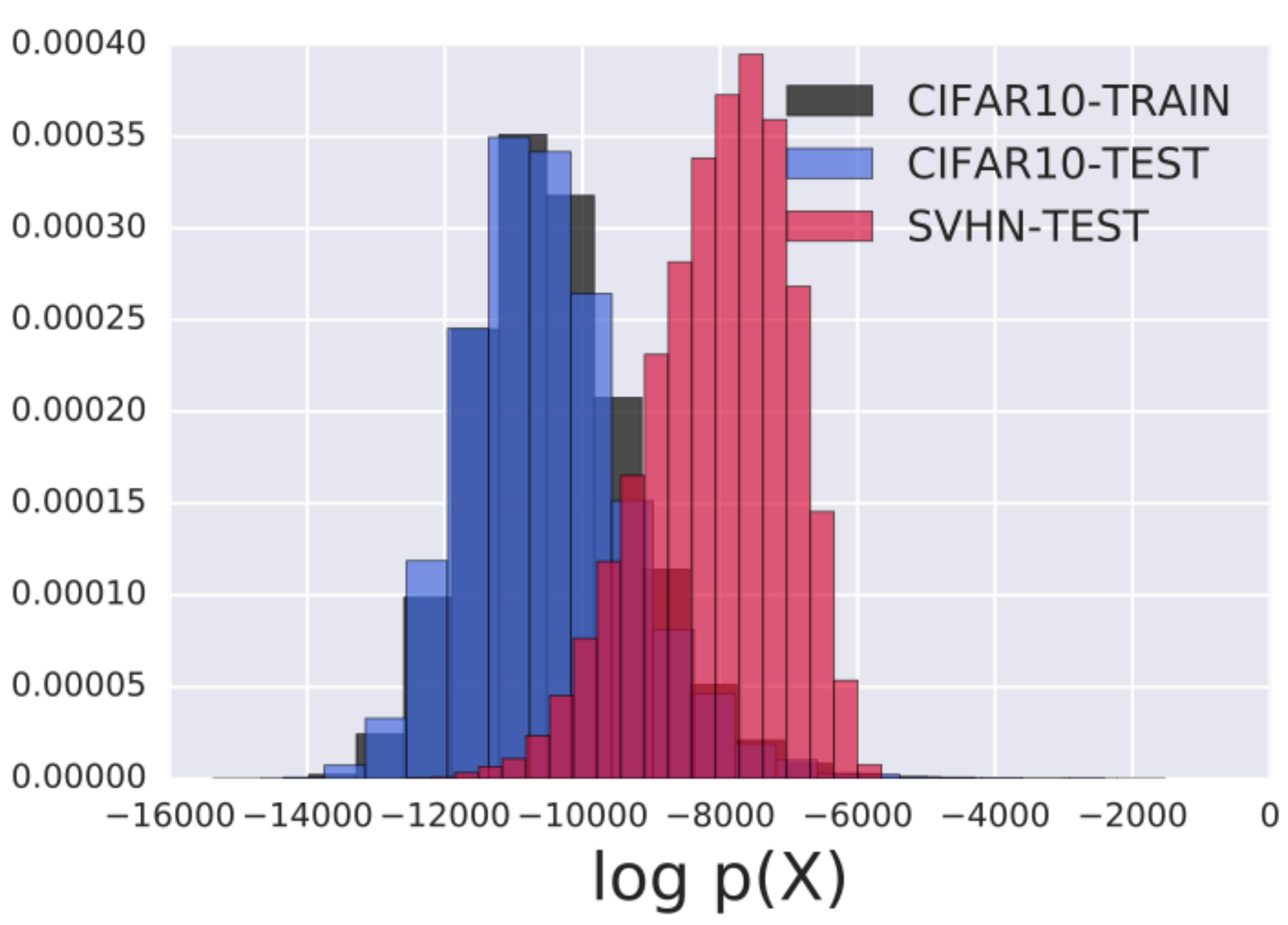
Recall earlier: overconfidence is not exclusive to discriminative models, but what if it's only about predictive values again?



Glow



PixelCNN

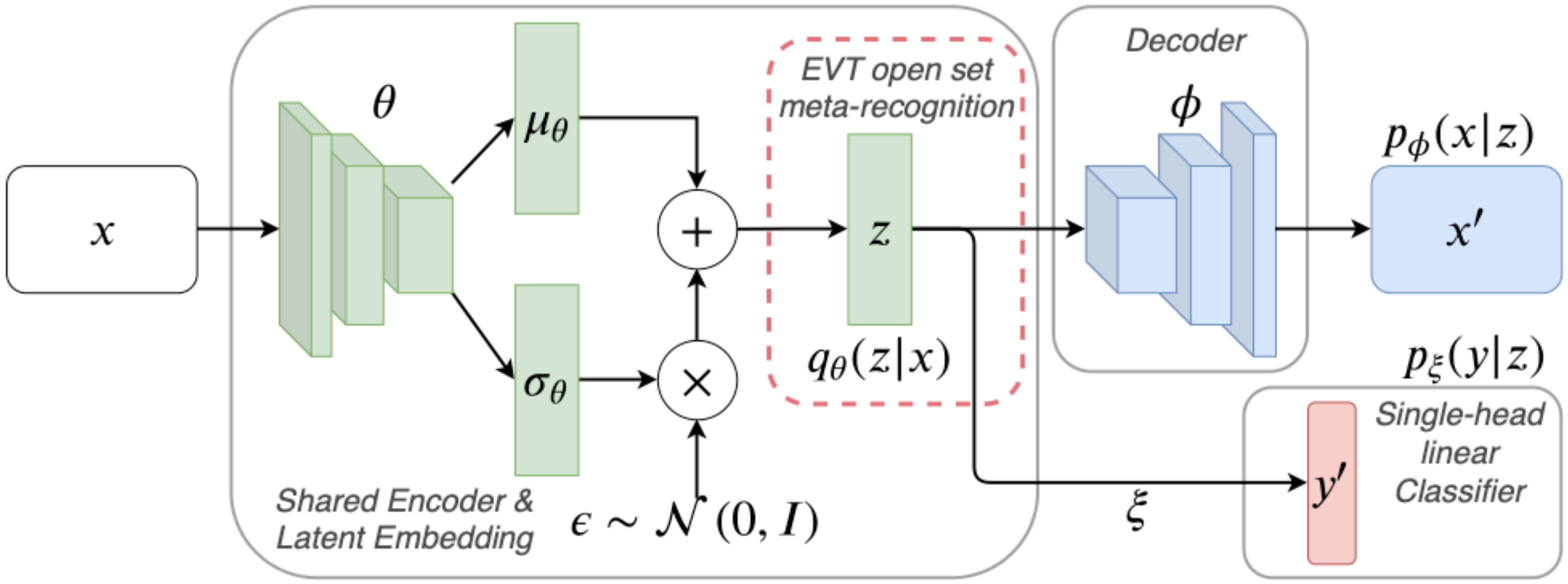


VAE

Nalisnick et al, "Do Deep Generative Models Know What They Don't Know", ICLR 2019

OpenMax in a generative variant

We could formulate an OpenMax variant based on a VAE, based on generative factors



Algorithm 1 Open set recognition calibration for deep variational neural networks. A Weibull model fit of tail-size η is conducted to bound the per class approximate posterior. Per class c Weibull models $\boldsymbol{\rho}_c$ with their respective shift τ_c, shape κ_c and scale λ_c parameters are returned.

Require: Trained encoder $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ and classifier $p_{\boldsymbol{\xi}}(\boldsymbol{y}|\boldsymbol{z})$

Require: Classifier probabilities $p_{\boldsymbol{\xi}}(\boldsymbol{y}|\boldsymbol{z})$ and samples from the approximate posterior $\boldsymbol{z}(\boldsymbol{x}^{(i)}) \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})$ for each training dataset example $\boldsymbol{x}^{(i)}$

Require: For each class c, let $\boldsymbol{S}_c^{(i)} = \boldsymbol{z}(\boldsymbol{x}_c'^{(i)})$ for each correctly classified training example $\boldsymbol{x}_c'^{(i)}$

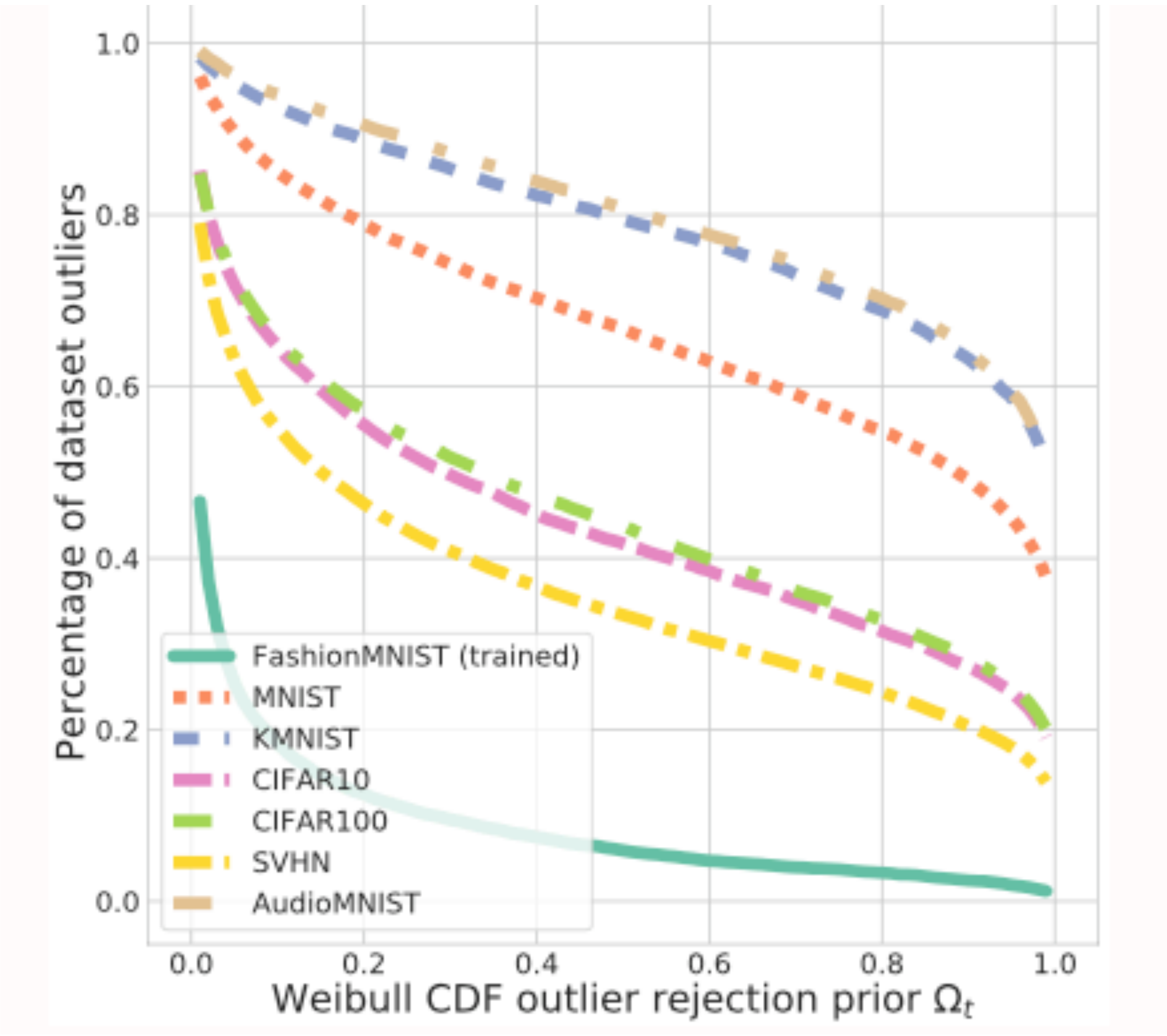
for $c = 1 \ldots C$ **do**
    Get per class latent mean $\bar{\boldsymbol{S}}_c = mean(\boldsymbol{S}_c^{(i)})$
    Weibull model $\boldsymbol{\rho}_c =$ Fit Weibull $\left(||\boldsymbol{S}_c - \bar{\boldsymbol{S}}_c||, \eta\right)$
Return means $\bar{\boldsymbol{S}}$ and Weibull models $\boldsymbol{\rho}$

Mundt et al "Open Set Recognition Through Deep Neural Network Uncertainty, Does Out-of-Distribution Detection Require Generative Classifiers?", ICCV Statistical Deep Learning Workshop 2019
& Mundt et al, "Unified Probabilistic Deep Continual Learning Through Open Set Recognition and Generative Replay", Journal of Imaging, Volume 8, Issue 4, 2022

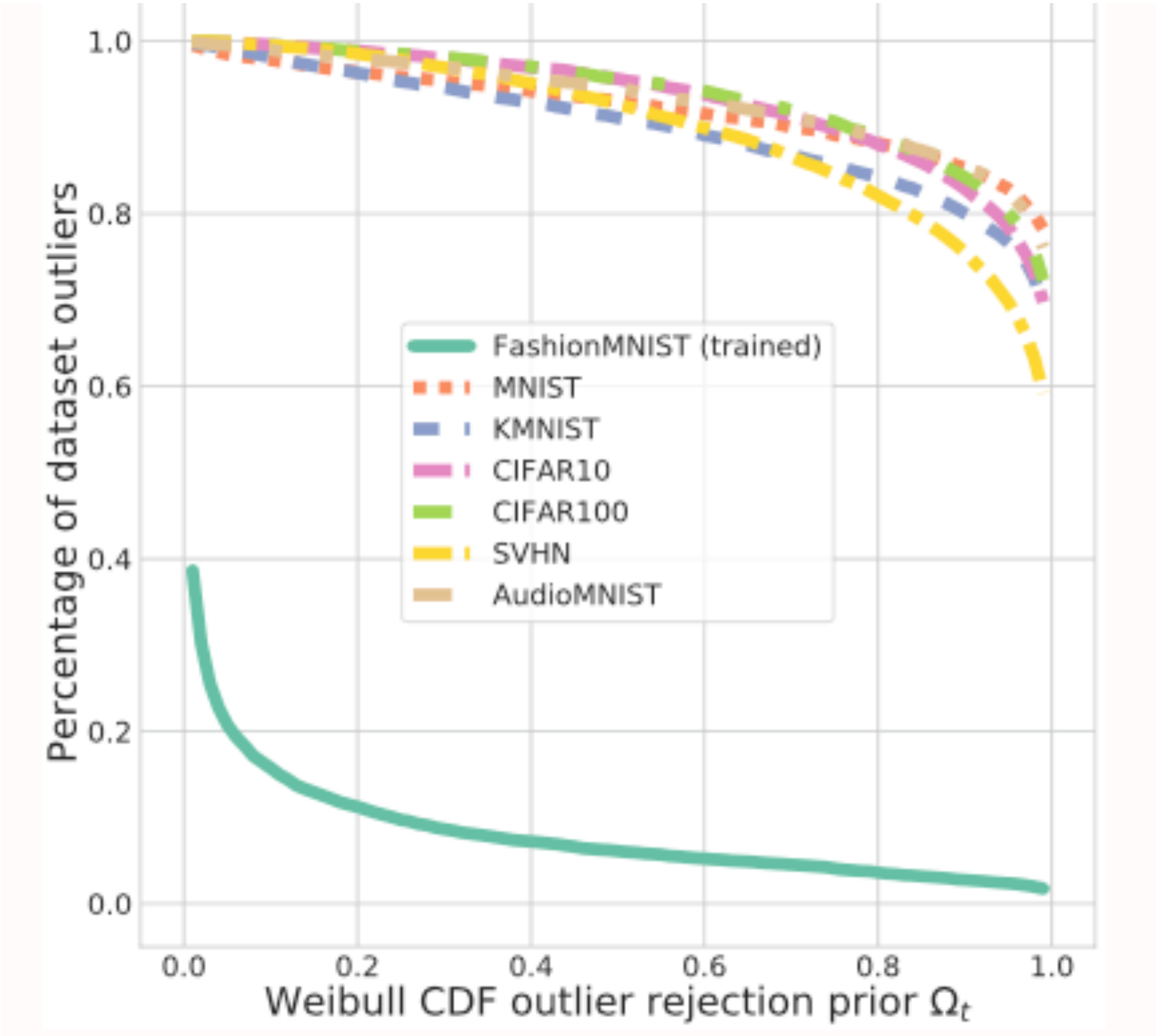
OpenMax in a generative variant

It may indeed be a question of the learned representations

Standard classifier p(y|x) with OpenMax



“Open”VAE approach: p(x,y)

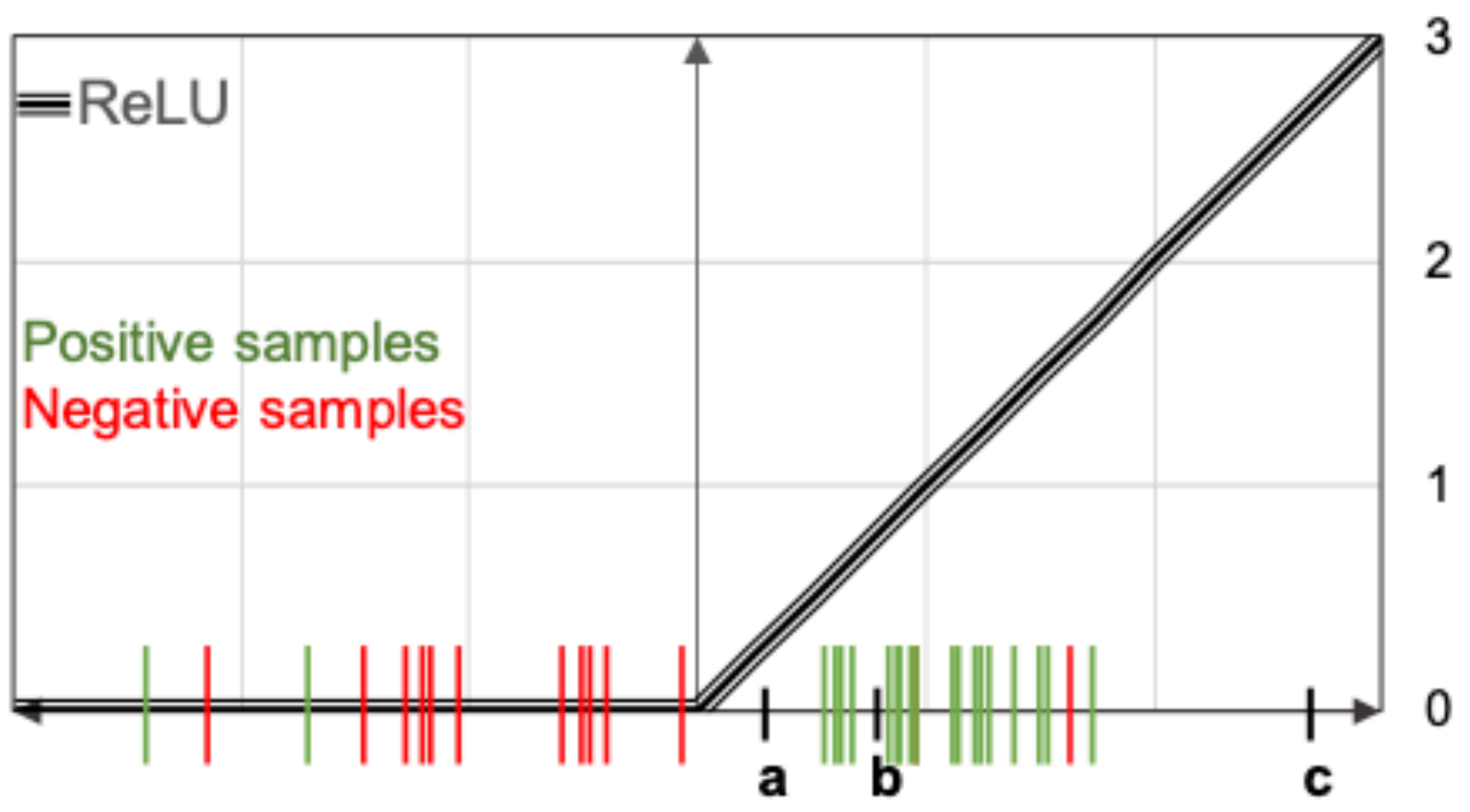


Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty, Does Out-of-Distribution Detection Require Generative Classifiers?”, ICCV Statistical Deep Learning Workshop 2019
& Mundt et al, “Unified Probabilistic Deep Continual Learning Through Open Set Recognition and Generative Replay”, Journal of Imaging, Volume 8, Issue 4, 2022

As an alternative/auxiliary approach, we could also take a direct look at the functions that we use in our model

An alternative/auxiliary view

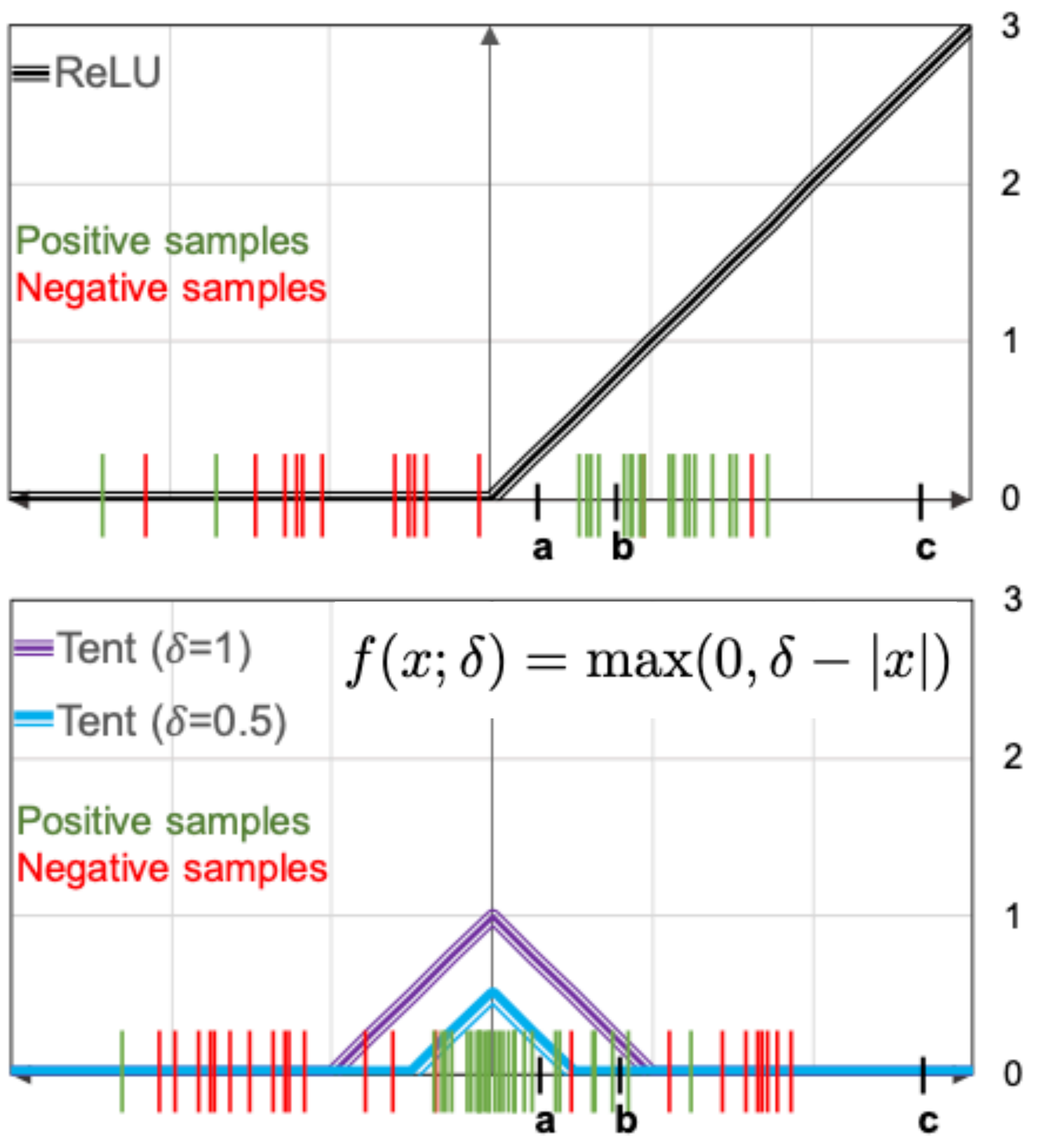
Hypothesis: specific functions in our ML models, like ReLU in NNs are (at least in parts) the culprit - they always produce high confidence far away from the data (Hein et al, “Why ReLU networks yield high confidence predictions far away from the training data and how to mitigate the problem”, CVPR 2019)



An alternative/auxiliary view

Hypothesis: specific functions in our ML models, like ReLU in NNs are (at least in parts) the culprit - they always produce high confidence far away from the data (Hein et al, “Why ReLU networks yield high confidence predictions far away from the training data and how to mitigate the problem”, CVPR 2019)

Alternative idea: use functions that are bounded and try to determine their “extent” based on the observed data (Rozsa & Boult, “Improved Adversarial Robustness by Reducing Open Space Risk via Tent Activations”, 2019)



An alternative/auxiliary view

Hypothesis: specific functions in our ML models, like ReLU in NNs are (at least in parts) the culprit - they always produce high confidence far away from the data (Hein et al, "Why ReLU networks yield high confidence predictions far away from the training data and how to mitigate the problem", CVPR 2019)

Alternative idea: use functions that are bounded and try to determine their "extent" based on the observed data (Rozsa & Boult, "Improved Adversarial Robustness by Reducing Open Space Risk via Tent Activations", 2019)

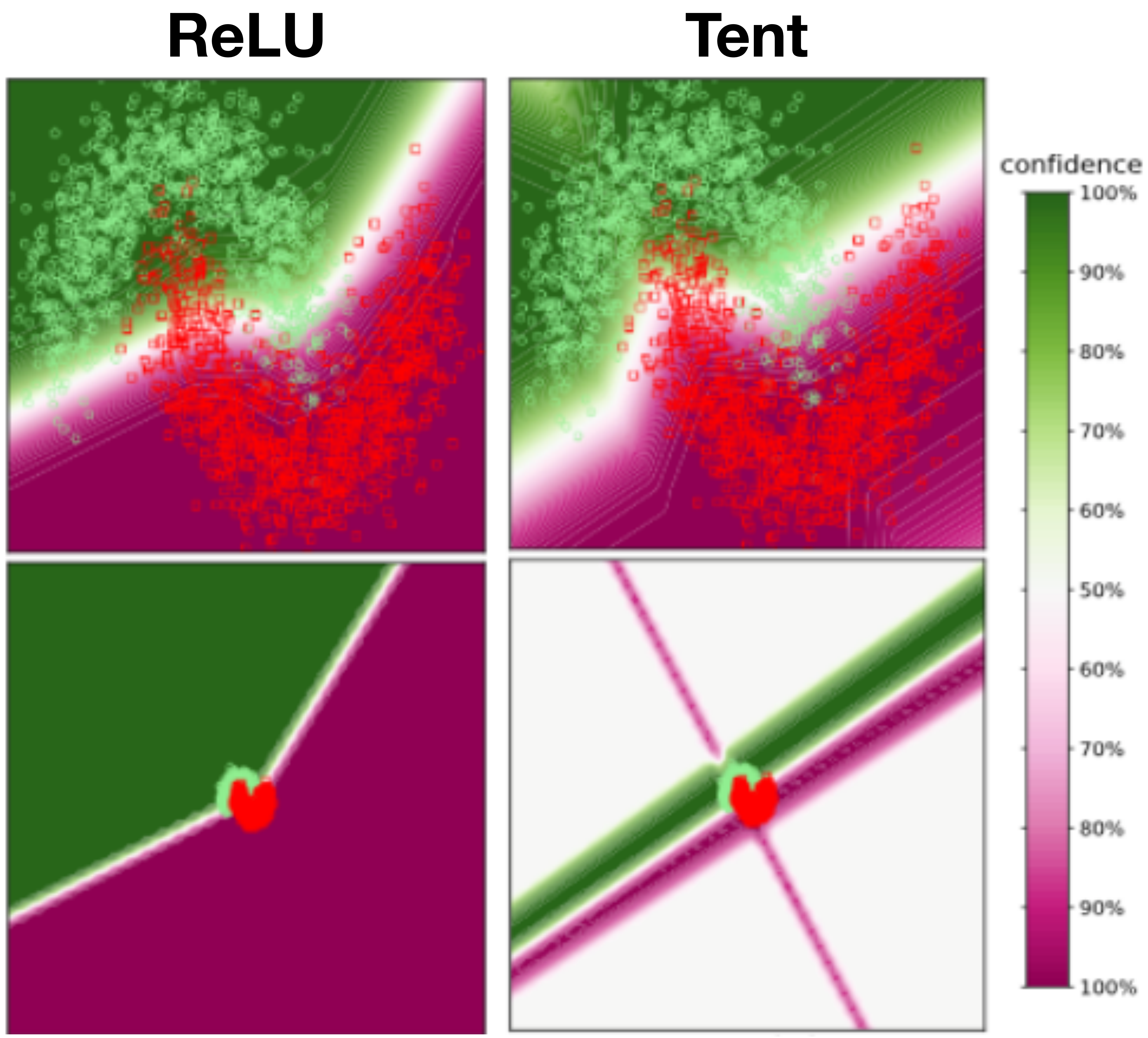


Figure produced by Quentin Delfosse, illustrating ReLU vs Tent activations

Open world learning: combining ideas

Open world learning

In retrospect: although there have been increments, the types of continual learning we have seen so far were indeed in a closed world

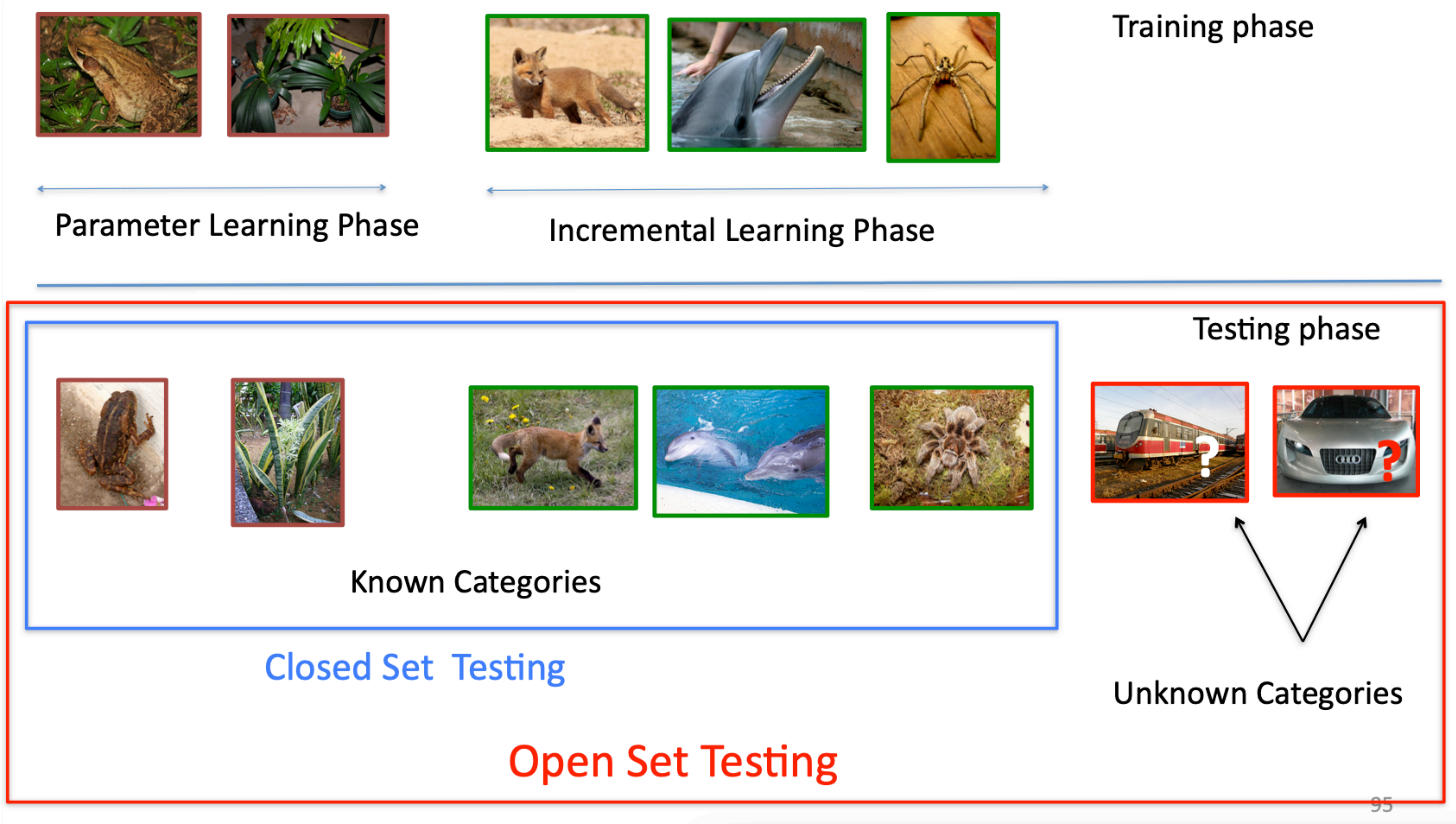
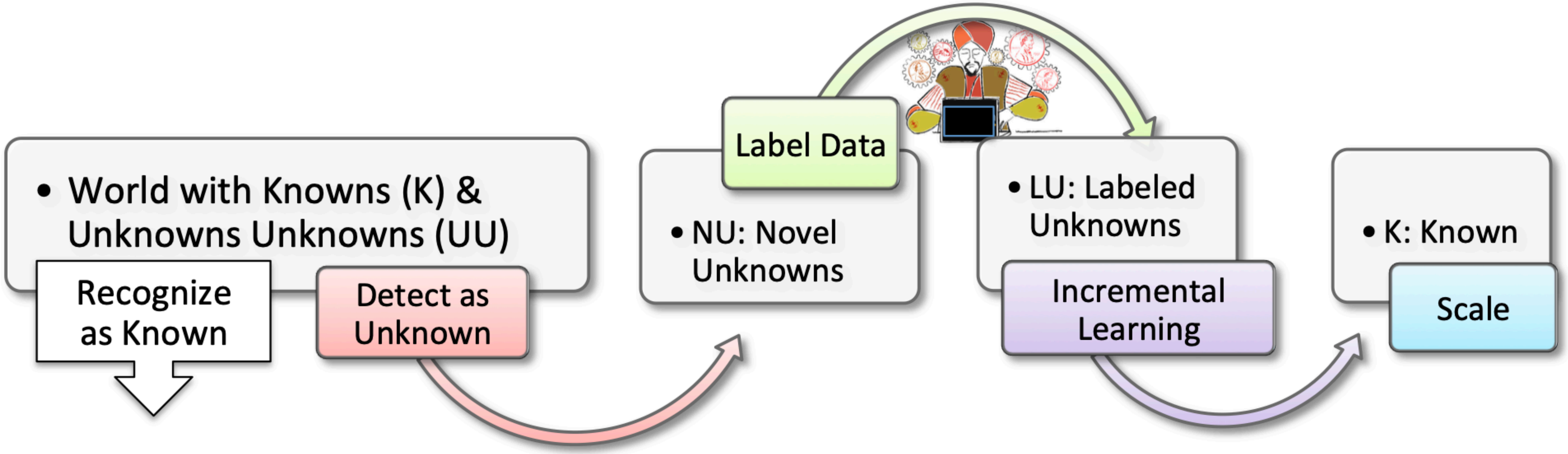


Figure from CVPR16 “Statistical Methods for Open Set Recognition” by Scheirer & Boult, https://www.wjscheirer.com/misc/openset/cvpr2016-open-set-part3.pdf

Open world learning

Open world learning tries to "puzzle together" the pieces we have seen so far

"An effective open world recognition system must efficiently perform four tasks: detect unknown, choose which points to label for addition to the model, label the points, and update the model" (Boult et al, "Learning and the Unknown", AAAI 2019)

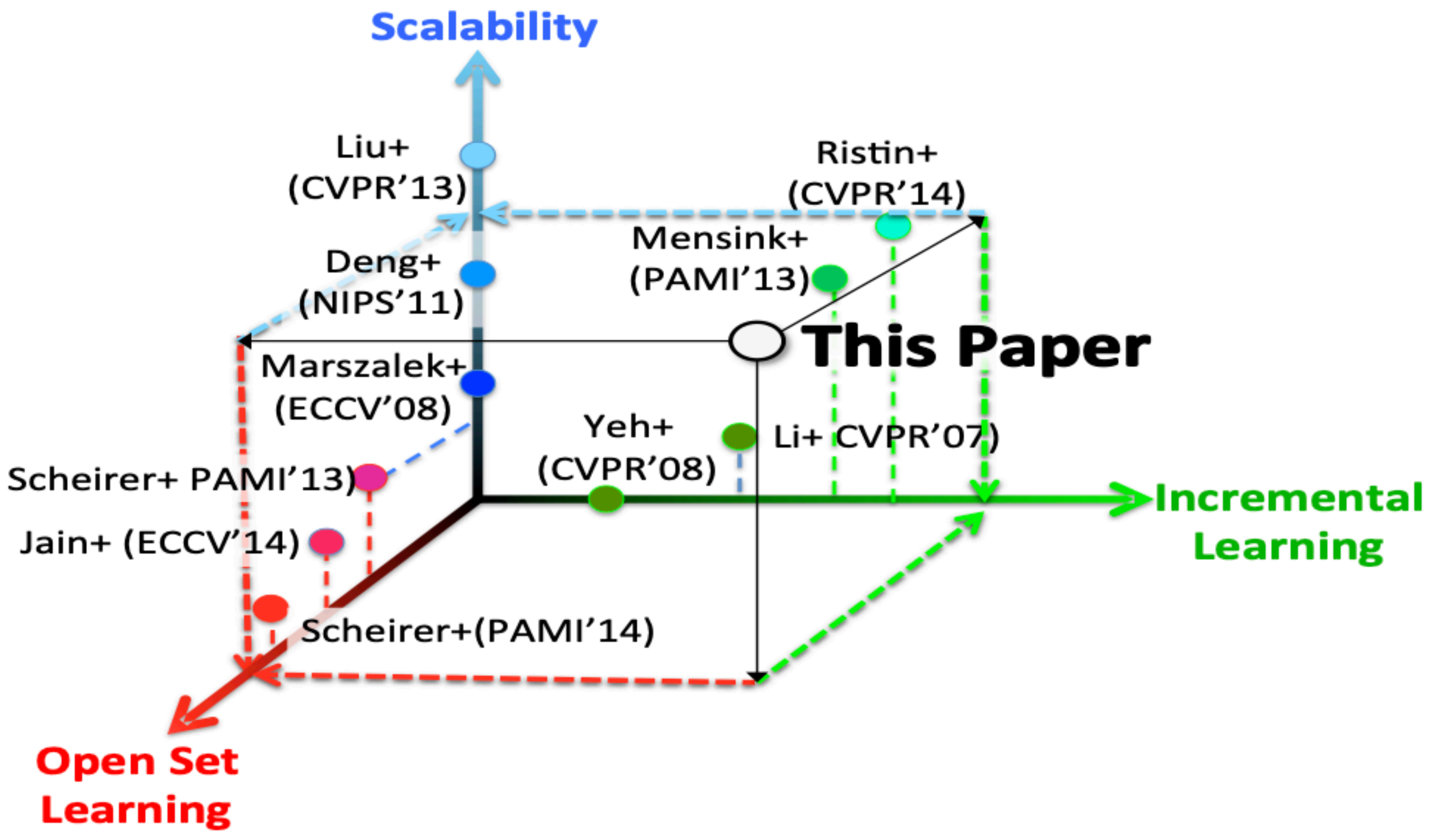


Bendale & Boult ,"Towards Open World Recognition", CVPR 2015

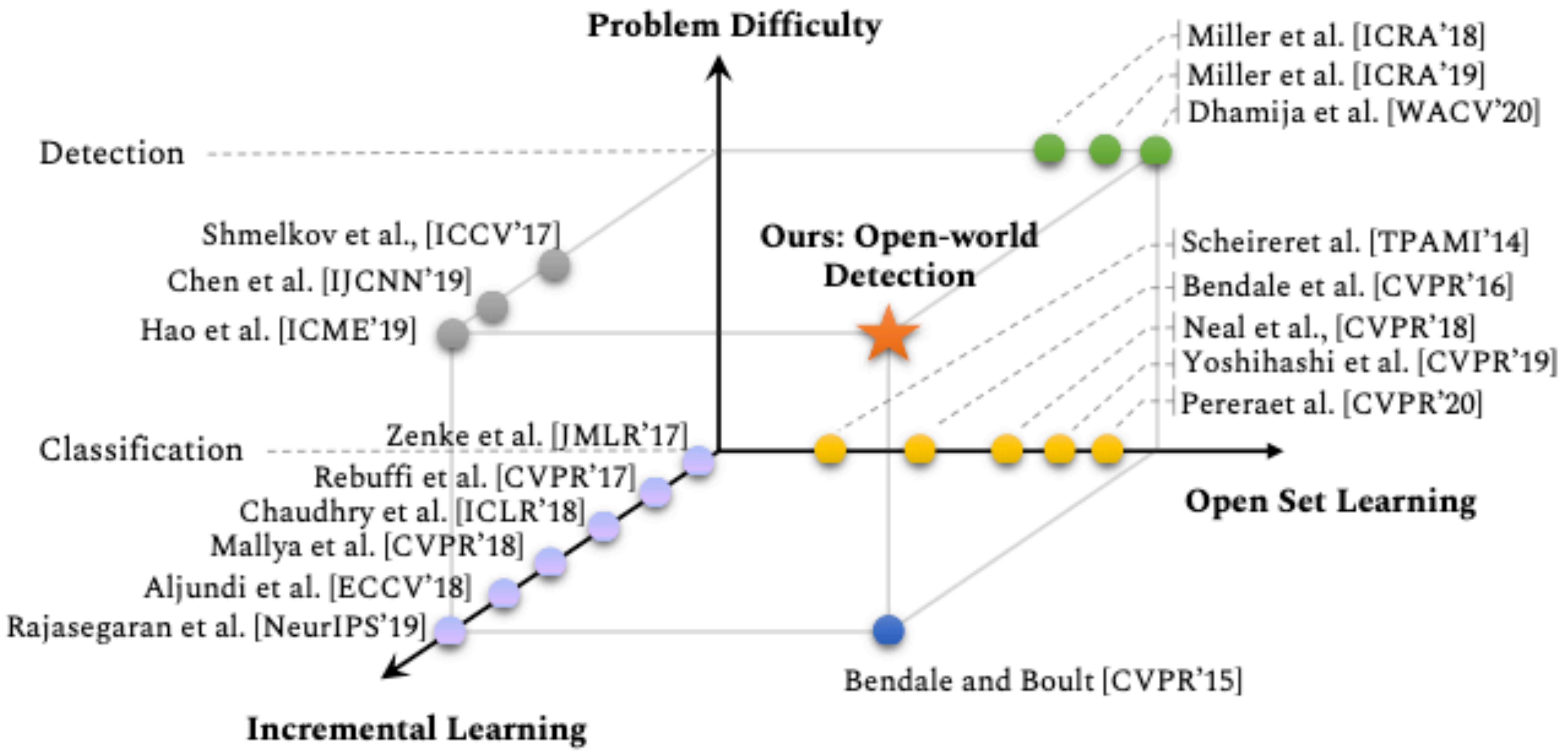
Open world learning

Open world learning tries to "puzzle together" the pieces we have seen so far

"An effective open world recognition system must efficiently perform four tasks: detect unknown, choose which points to label for addition to the model, label the points, and update the model" (Boult et al, "Learning and the Unknown", AAAI 2019)



Bendale & Boult ,"Towards Open World Recognition", CVPR 2015



Joseph et al, "Towards Open World Object Detection", CVPR 2021

Open world learning

We can try to puzzle the pieces together now. As it is very much a cutting-edge research frontier, let's talk about it more in the "frontiers" lecture

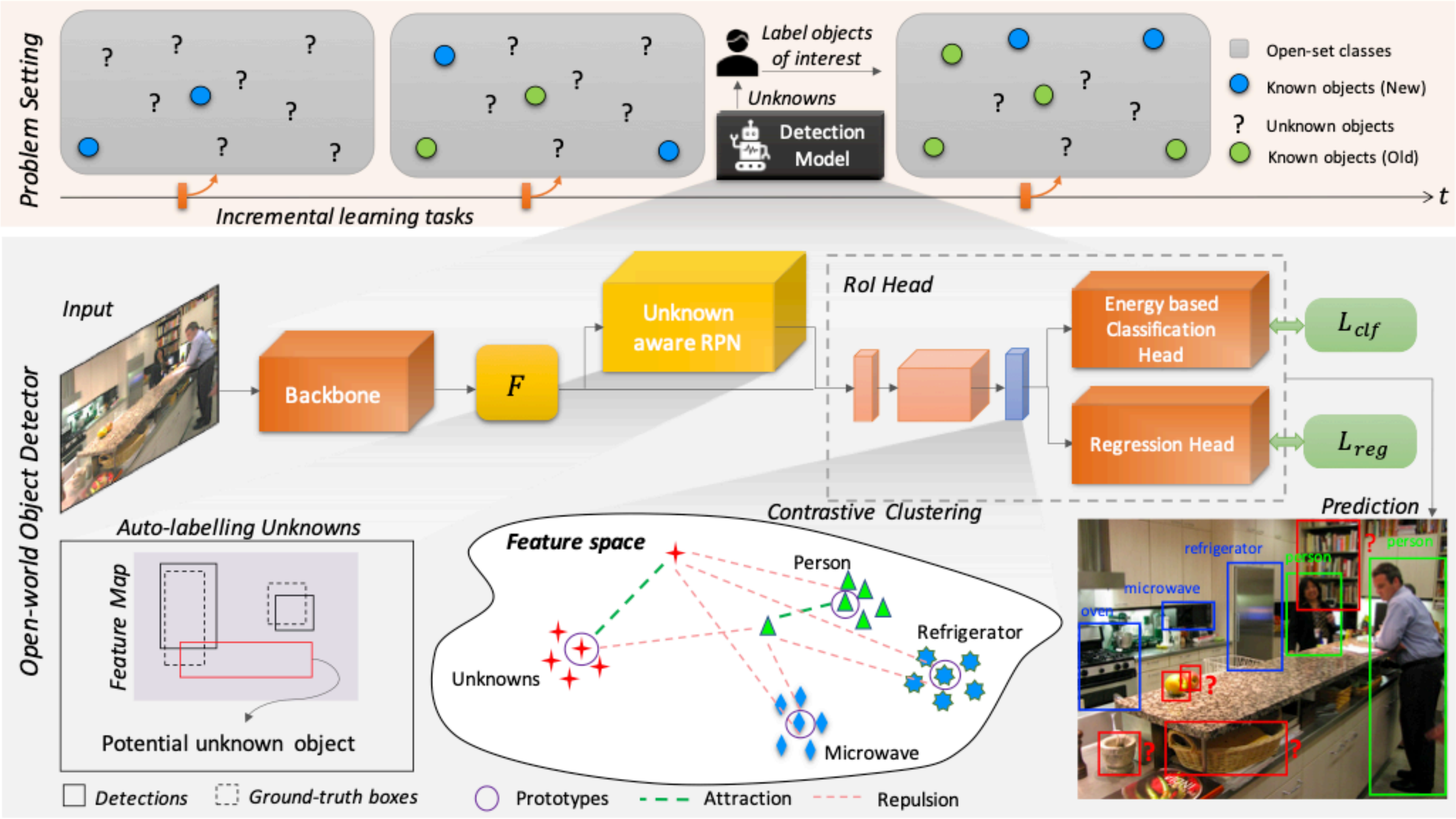


Figure 2: *Approach Overview: Top row:* At each incremental learning step, the model identifies unknown objects (denoted by '?'), which are progressively labelled (as blue circles) and added to the existing knowledge base (green circles). *Bottom row:* Our open world object detection model identifies potential unknown objects using an energy-based classification head and the unknown-aware RPN. Further, we perform contrastive learning in the feature space to learn discriminative clusters and can flexibly add new classes in a continual manner without forgetting the previous classes.

Joseph et al, "Towards Open World Object Detection", CVPR 2021

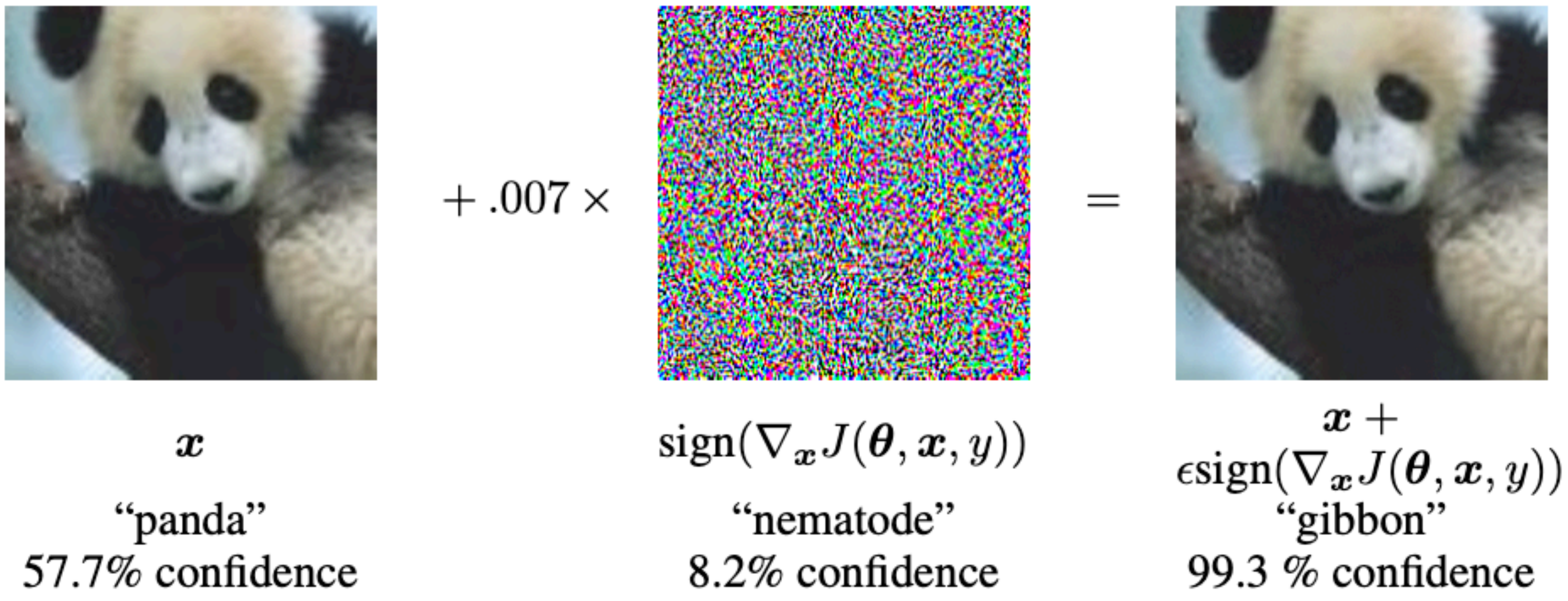
Frontiers

Ending on some open questions & a disclaimer:

- Note the “towards” in many of the paper titles
- There is much to be done still: what about avoiding forgetting in addition now?
- Naturally, evaluation gets even more complicated now!
- It’s no longer a question of ML algorithms, perhaps it already was a systems question beforehand, but now it definitely is

Corruptions, adversarial etc.

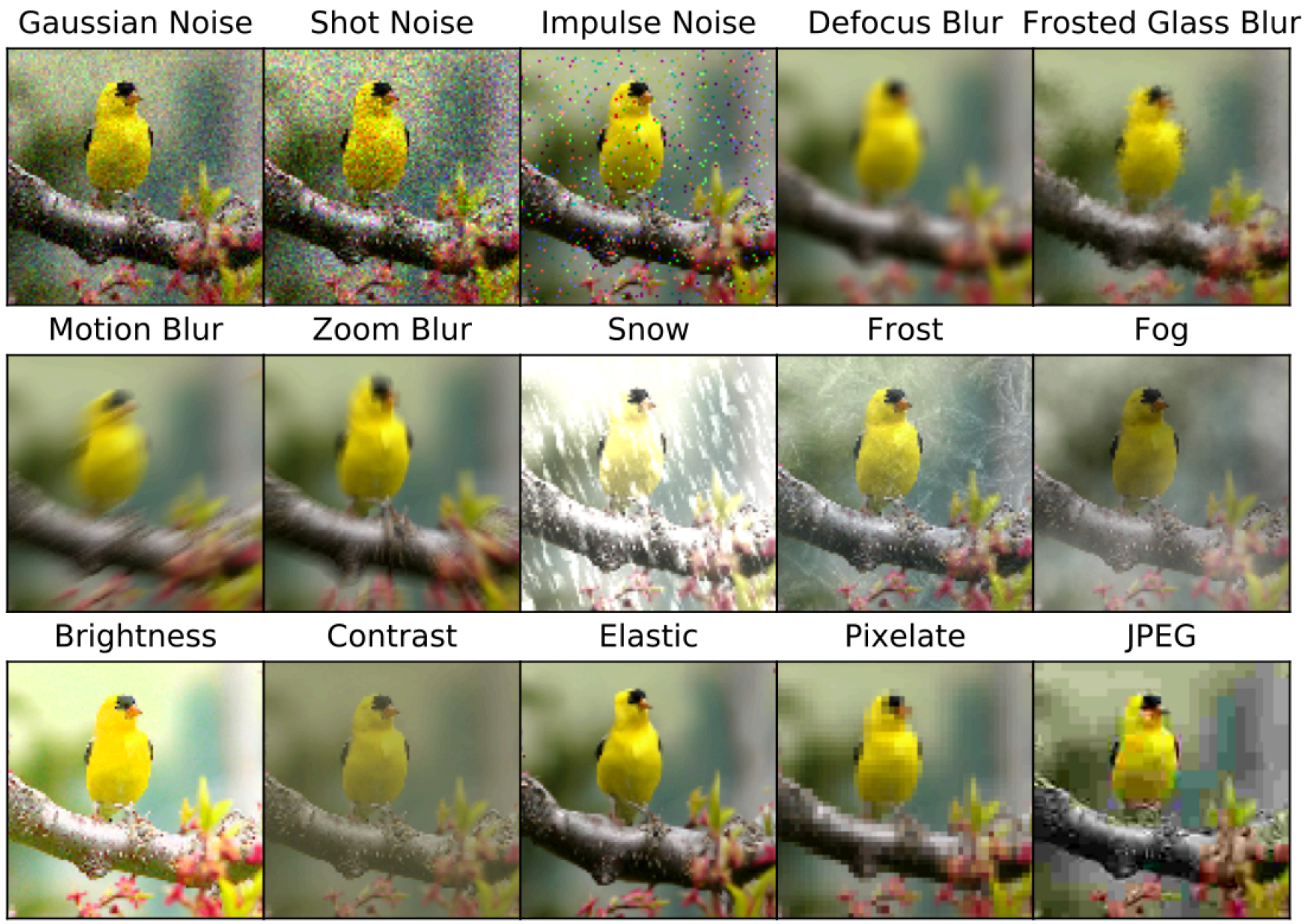
What about natural corruptions, adversarial attacks etc.?



Goodfellow et al, "Explaining and Harnessing Adversarial Examples", ICLR 2015

Corruptions, adversarial etc.

What about natural corruptions, adversarial attacks etc.?



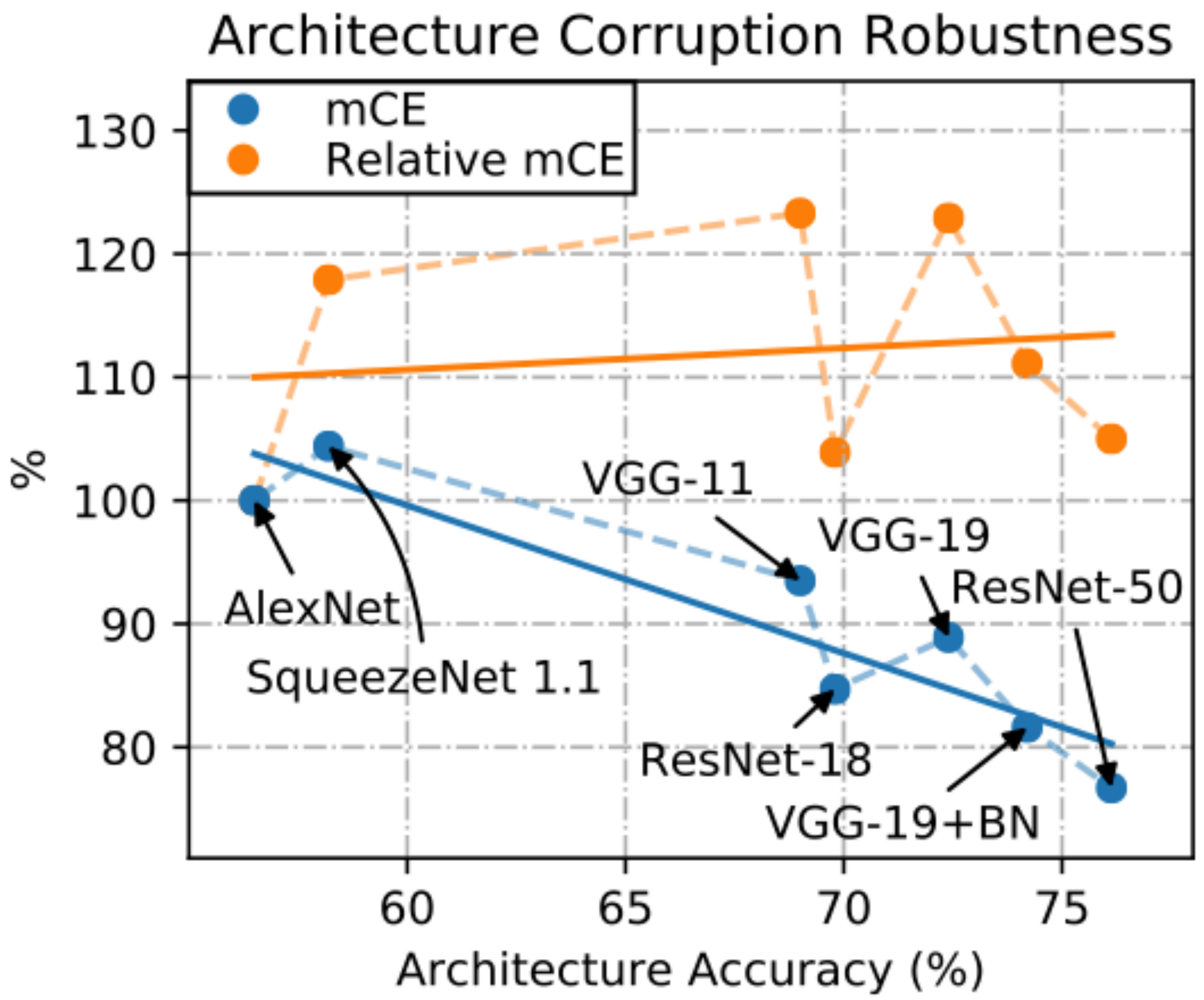


Figure 3: Robustness (mCE) and Relative mCE IMAGENET-C values. Relative mCE values suggest robustness in itself declined from AlexNet to ResNet. "BN" abbreviates Batch Normalization.

Hendricks & Dietterich, "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations", ICLR 2019