Outline

- Introduction
  - Predictive uncertainty of deep neural networks
  - Summary of contributions
- How to train confident neural networks
  - Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples [Lee’ 18a]
- Applications
  - Hierarchical novelty detection [Lee’ 18b]
- Conclusion
  - Future work

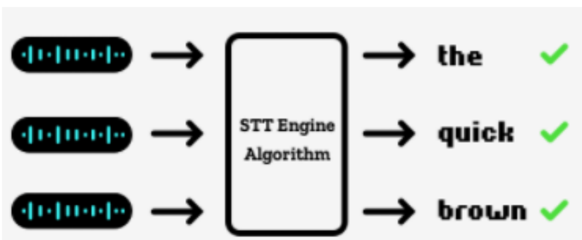
[Lee’ 18a] Lee, K., Lee, H., Lee, K. and Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. I n ICLR, 2018.
[Lee’ 18b] Lee, K., Lee, Min. K, Zhang, Y. Shin. J, Lee, H. Hierarchical Novelty Detection for Visual Object Recognition, In CVPR, 2018.

Introduction: Predictive uncertainty of deep neural networks (DNNs)

- Supervised learning (e.g., regression and classification)
  - Objective: finding an unknown target distribution, i.e., P(Y|X)

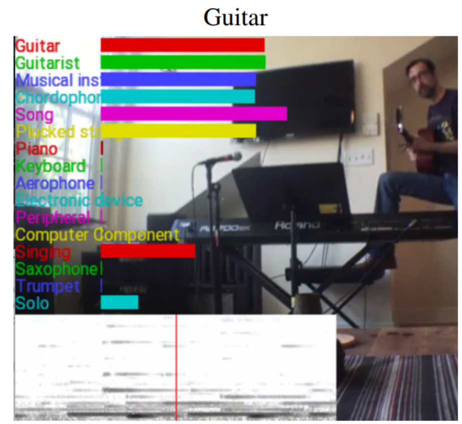
$$\text{Input space } \textcircled{X} \xrightarrow{P} \textcircled{Y} \text{ Output space}$$

  - Recent advances in deep learning have dramatically improved accuracy on several supervised learning tasks

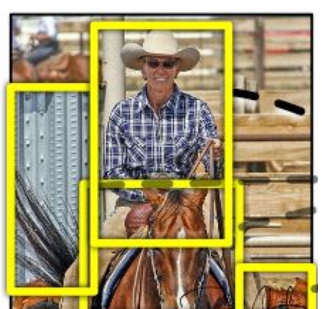


Speech recognition [Amodei' 16]

Image classification [He' 16]



Audio recognition [Hershey' 17]

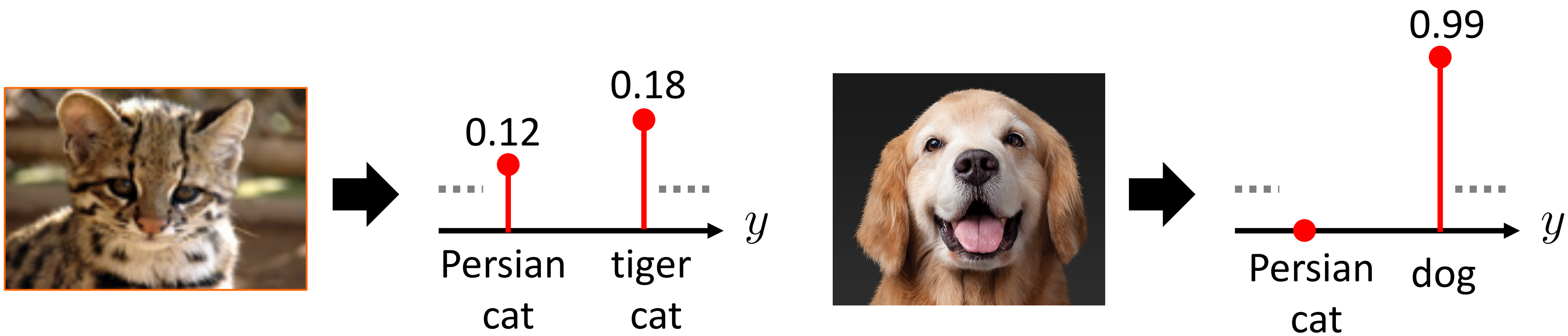


Objective detection [Girshick' 15]

[Amodei' 16] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G. and Chen, J. Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML, 2016.*
[He' 16] He, K., Zhang, X., Ren, S. and Sun, J. Deep residual learning for image recognition. In *CVPR, 2016.*
[Hershey' 17] Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B. and Slaney, M. CNN architectures for large-scale audio classification. In *ICASSP, 2017.*
[Girshick' 15] Girshick, Ross. Fast r-cnn. In ICCV, pp. 1440–1448, 2015

Introduction: Predictive uncertainty of deep neural networks (DNNs)

- Uncertainty of predictive distribution is important in DNN's applications
  - What is predictive uncertainty?
    - As a example, consider classification task



    - It represents a confidence about prediction!
  - For example, it can be measured as follows:
    - Entropy of predictive distribution [Lakshminarayanan' 17]

$$\sum_y -P(y|\mathbf{x}) \log P(y|\mathbf{x})$$

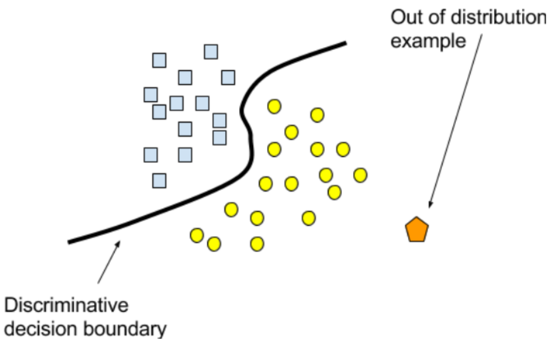
    - Maximum value of predictive distribution [Hendrycks' 17]

$$\max_y P(y|\mathbf{x})$$

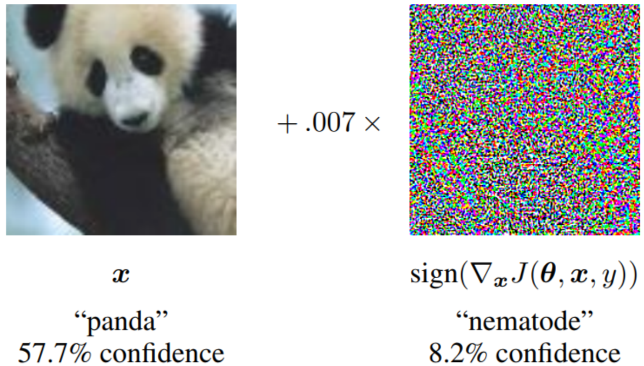
[Lakshminarayanan' 17] Lakshminarayanan, B., Pritzel, A. and Blundell, C., Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS, 2017.*
[Henderycks' 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*

Introduction: Predictive uncertainty of deep neural networks (DNNs)

- Predictive uncertainty is related to many machine learning problems:



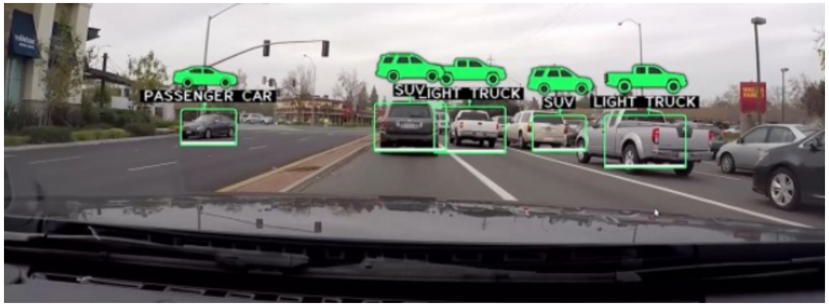
Novelty detection [Hendrycks' 17]



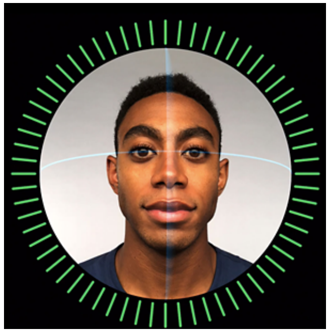
Adversarial detection [Song' 18]

Ensemble learning [Lee' 17]

- Predictive uncertainty is also indispensable when deploying DNNs in real-world systems [Dario' 16]



Autonomous drive

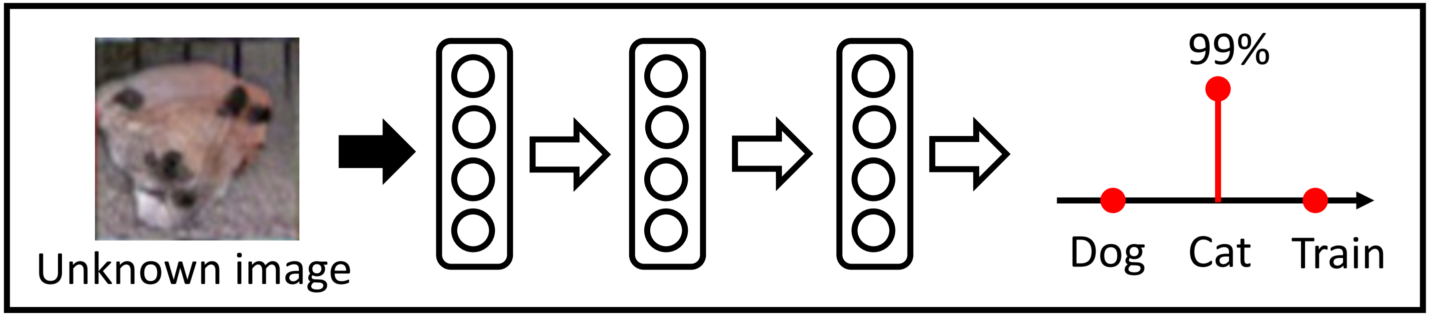


Secure authentication system

[Dario' 16] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.
[Henderycks' 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*
[Guo' 17] Guo, C., Pleiss, G., Sun, Y. and Weinberger, K.Q., 2017. On Calibration of Modern Neural Networks. *In ICML 2017.*
[Goodfellow' 14] Goodfellow, I.J., Shlens, J. and Szegedy, C., 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
[Srivastava' 14] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., Dropout: a simple way to prevent neural networks from overfitting. JMLR. 2014.

Introduction: Predictive uncertainty of deep neural networks (DNNs)

- However, DNNs do not capture their predictive uncertainty



  - E.g., DNNs trained to classify MNIST images often produce high confident probability 91% even for random noise [Henderycks' 17]
  - Challenge arises in improving the quality of the predictive uncertainty!

- Main topic of this presentation
  - How to train confident neural networks?
    - Training confidence-calibrated classifiers for detecting out-of-distribution samples [Lee' 18a]
  - Applications
    - Confident multiple choice learning [Lee' 17]
    - Hierarchical novelty detection [Lee' 18b]

[Henderycks' 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*
[Lee' 18a] Lee, K., Lee, H., Lee, K. and Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. In ICLR 2018.
[Lee' 17] Lee, K., Hwang, C., Park, K. and Shin, J. Confident Multiple Choice Learning. *In ICML, 2017.*
[Lee' 18b] Lee, K., Lee, Min. K, Zhang, Y. Shin. J, Lee, H. Hierarchical Novelty Detection for Visual Object Recognition, In CVPR, 2018.

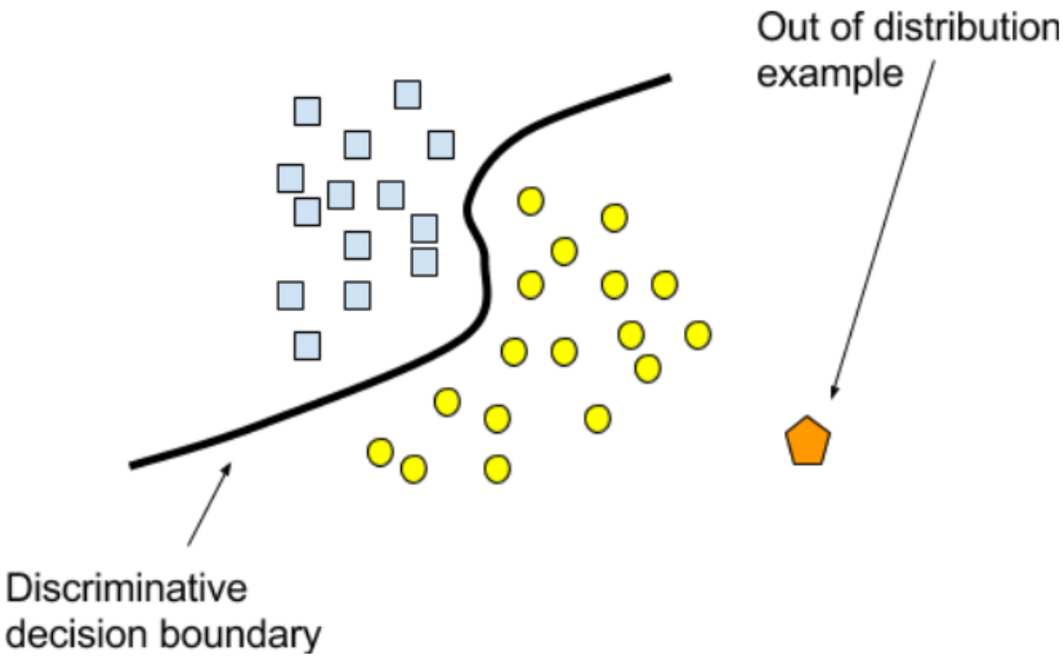
Outline

- Introduction
  - Predictive uncertainty of deep neural networks
  - Summary of contributions
- How to train confident neural networks
  - Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples [Lee' 18a]
- Applications
  - Confident Multiple Choice Learning [Lee' 17]
  - Hierarchical novelty detection [Lee' 18b]
- Conclusion
  - Future work

[Lee' 18a] Lee, K., Lee, H., Lee, K. and Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. In ICLR, 2018.
[Lee' 17] Lee, K., Hwang, C., Park, K. and Shin, J. Confident Multiple Choice Learning. *In ICML, 2017.*
[Lee' 18b] Lee, K., Lee, Min. K, Zhang, Y. Shin. J, Lee, H. Hierarchical Novelty Detection for Visual Object Recognition, In CVPR, 2018.

How to Train Confident Neural Networks?

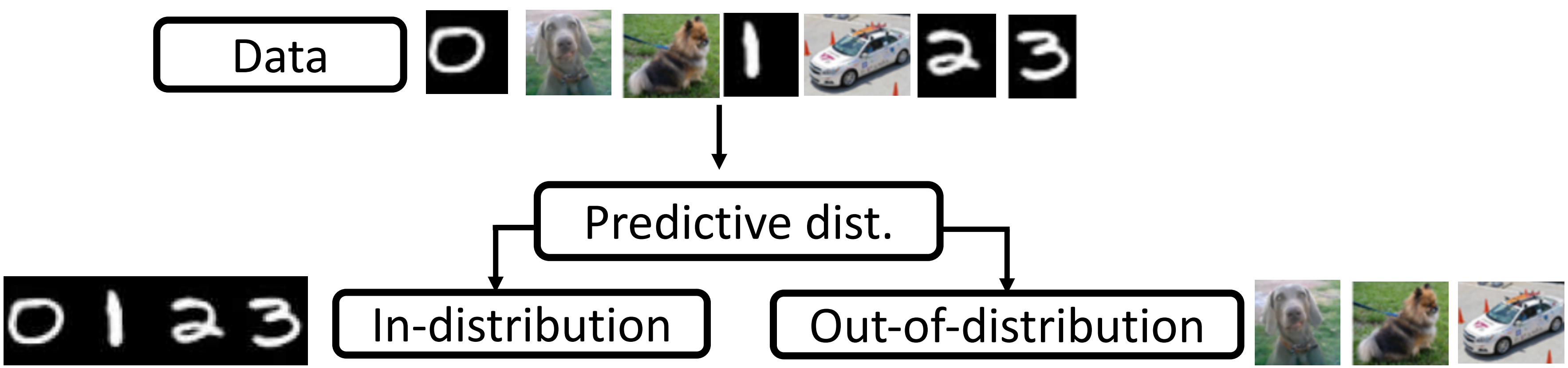
- Related problem
  - Detecting out-of-distribution [Hendrycks' 17, Liang' 18]
    - Detect whether a test sample is from in-distribution (i.e., training distribution by classifier) or out-of-distribution



[Henderycks' 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*
[Liang' 18] Liang, S., Li, Y. and Srikant, R. Principled Detection of Out-of-Distribution Examples in Neural Networks. *In ICLR, 2018.*

How to Train Confident Neural Networks?

- Related problem
  - Detecting out-of-distribution [Hendrycks’ 17, Liang’ 18]
    - Detect whether a test sample is from in-distribution (i.e., training distribution by classifier) or out-of-distribution
  - E.g., image classification
    - Assume a classifier trains handwritten digits (denoted as in-distribution)
    - Detecting out-of-distribution

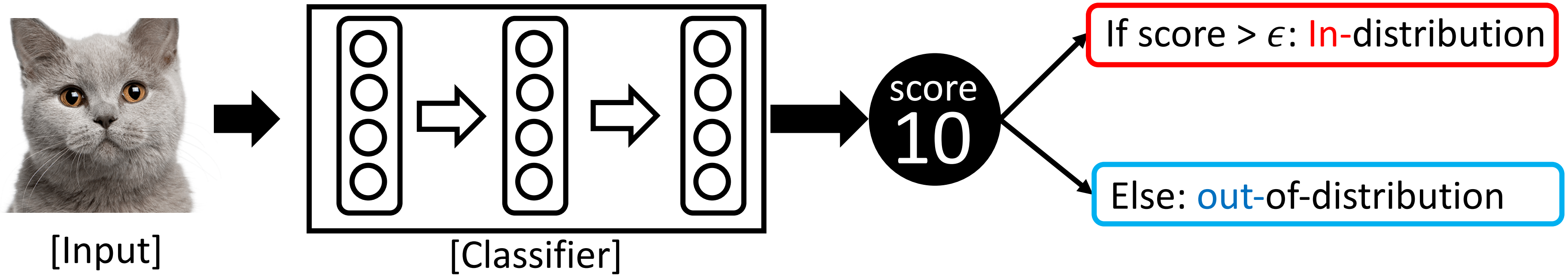


  - Performance of detector reflects confidence of predictive distribution!

[Henderycks’ 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*
[Liang’ 18] Liang, S., Li, Y. and Srikant, R. Principled Detection of Out-of-Distribution Examples in Neural Networks. *In ICLR, 2018.*

Related Work

- Threshold-based Detector [Guo’ 17, Hendrycks’17, Liang’ 18]



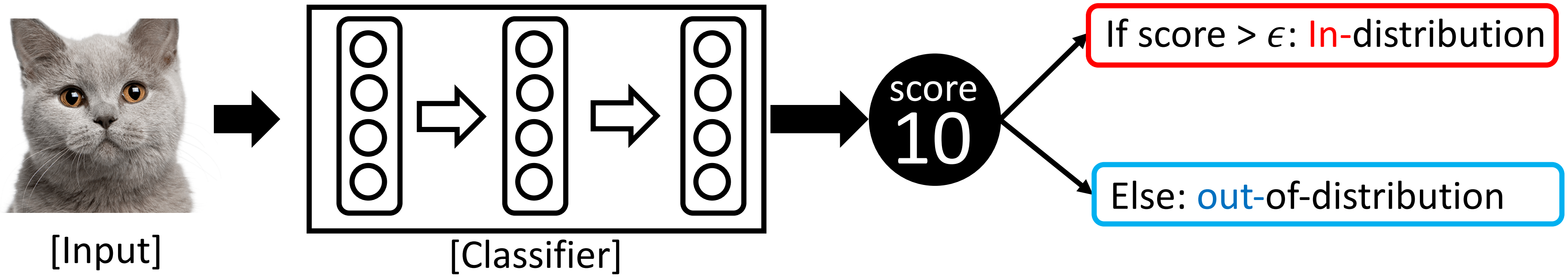
[Henderycks’ 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*

[Guo’ 17] Guo, C., Pleiss, G., Sun, Y. and Weinberger, K.Q., 2017. On Calibration of Modern Neural Networks. *In ICML 2017.*

[Liang’ 18] Liang, S., Li, Y. and Srikant, R., 2017. Principled Detection of Out-of-Distribution Examples in Neural Networks. *In ICLR, 2018.*

Related Work

- Threshold-based Detector [Guo’ 17, Hendrycks’17, Liang’ 18]



- How to define the score?
  - Baseline detector [Hendrycks’17]
    - Confidence score = maximum value of predictive distribution: $\max_y P(y|\mathbf{x})$
  - Temperature scaling [Guo’ 17]
    - Confidence score = maximum value of scaled predictive distribution

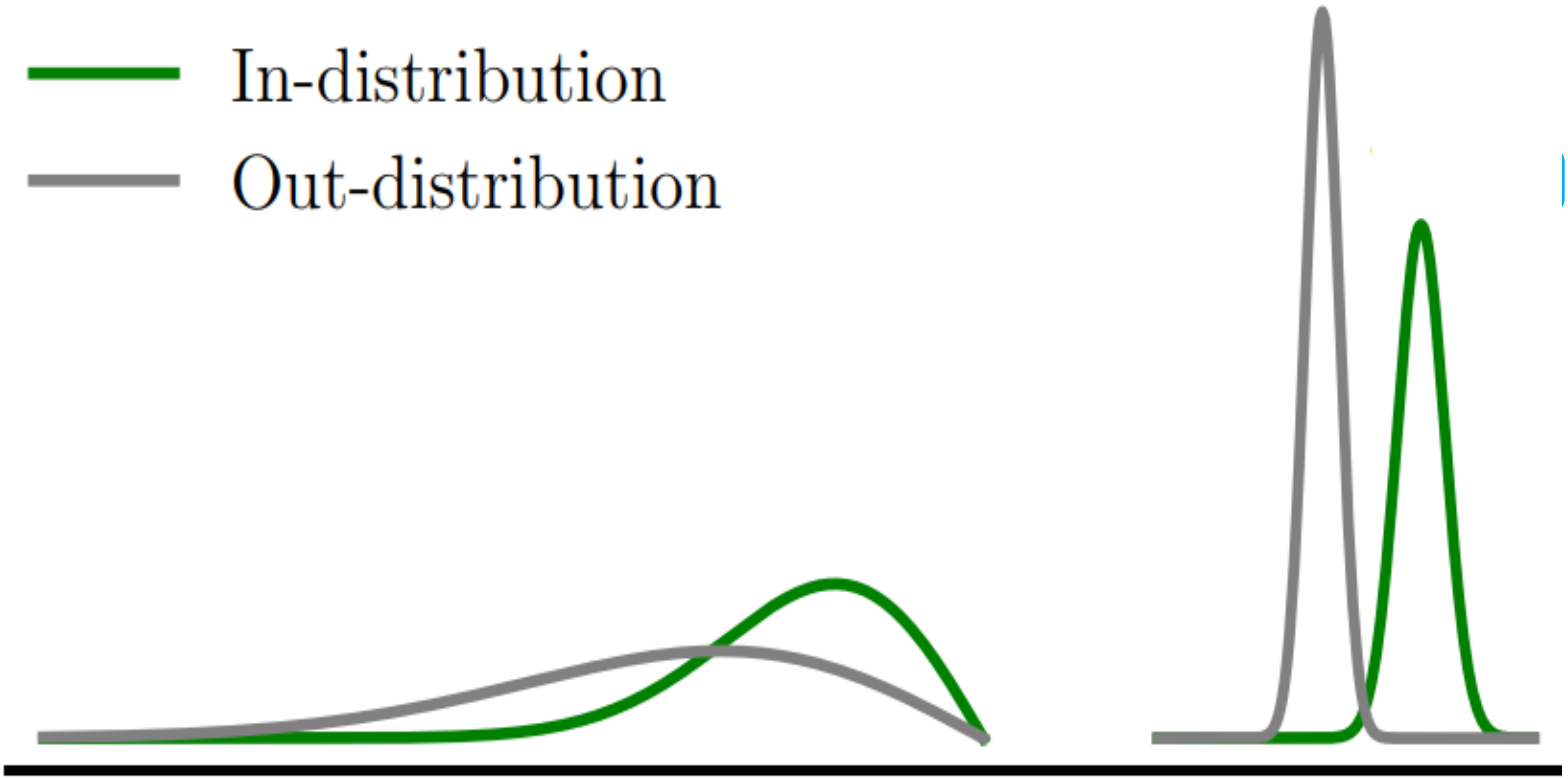
$$p_i(\boldsymbol{x}; T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^{N} \exp\left(f_j(\boldsymbol{x})/T\right)}$$

Output of neural networks

[Henderycks’ 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*
[Guo’ 17] Guo, C., Pleiss, G., Sun, Y. and Weinberger, K.Q., 2017. On Calibration of Modern Neural Networks. *In ICML 2017.*
[Liang’ 18] Liang, S., Li, Y. and Srikant, R., 2017. Principled Detection of Out-of-Distribution Examples in Neural Networks. *In ICLR, 2018.*

Related Work

- Threshold-based Detector [Guo’ 17, Hendrycks’17, Liang’ 18]



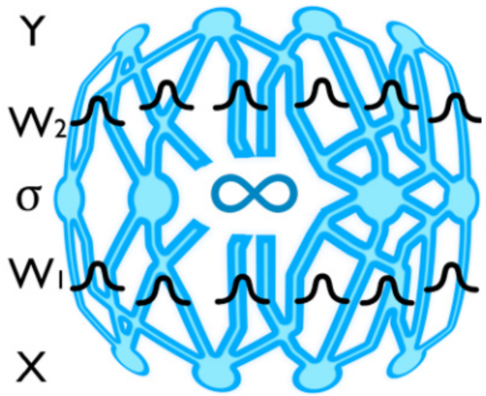
- Limitations
  - Performance of prior works highly depends on how to train the classifiers

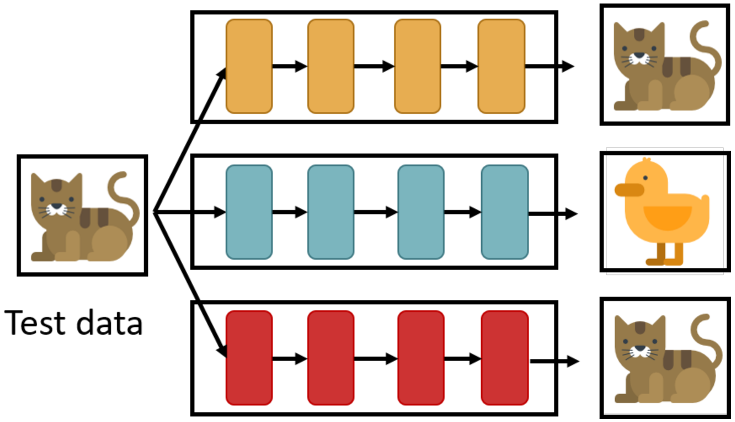
[Henderycks’ 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*
[Guo’ 17] Guo, C., Pleiss, G., Sun, Y. and Weinberger, K.Q., 2017. On Calibration of Modern Neural Networks. *In ICML 2017.*
[Liang’ 18] Liang, S., Li, Y. and Srikant, R., 2017. Principled Detection of Out-of-Distribution Examples in Neural Networks. *In ICLR, 2018.*

Our Contributions

[Yingzhen' 17] Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. In International Conference on Machine Learning (ICML), 2017.
[Balaji' 17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in neural information processing systems (NIPS), 2017.

- One can consider
  - Bayesian neural networks [Yingzhen' 17]
  - Ensemble of classifiers [Balaji' 17]



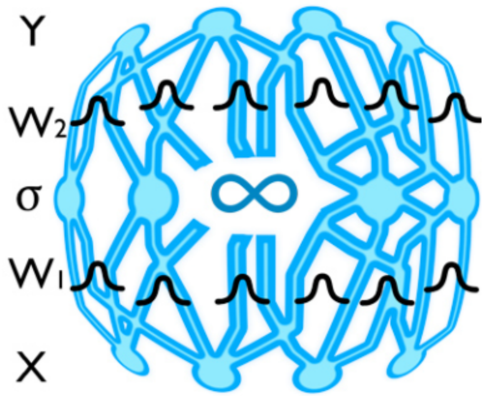


  - Training or inferring those models are computationally expensive

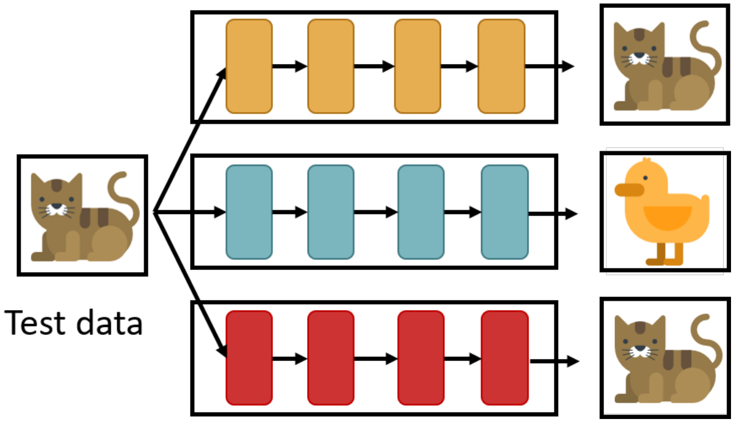
Our Contributions

[Yingzhen' 17] Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. In International Conference on Machine Learning (ICML), 2017.
[Balaji' 17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in neural information processing systems (NIPS), 2017.

- One can consider
  - Bayesian neural networks [Yingzhen' 17]



  - Ensemble of classifiers [Balaji' 17]



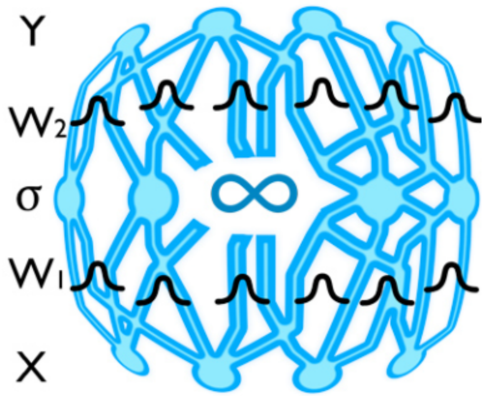
  - Training or inferring those models are computationally expensive
- Our contribution

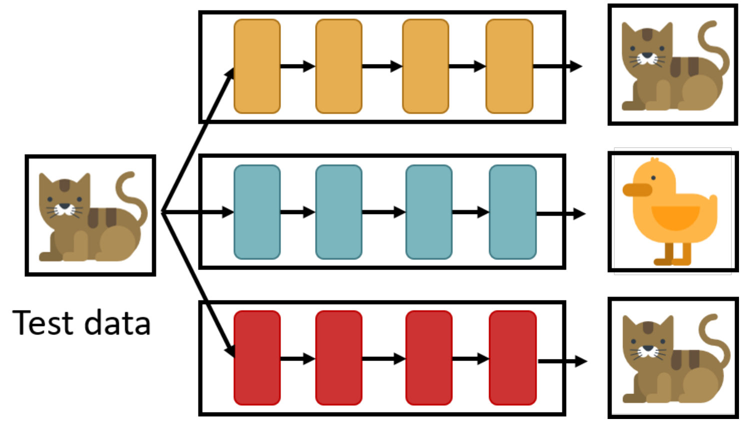
Confidence loss for training more plausible simple DNNs	GAN for generating out-of-distribution samples	Joint training method of classifier and GAN

Our Contributions

[Yingzhen' 17] Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. In International Conference on Machine Learning (ICML), 2017.
[Balaji' 17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in neural information processing systems (NIPS), 2017.

- One can consider
  - Bayesian neural networks [Yingzhen' 17]
  - Ensemble of classifiers [Balaji' 17]





  - Training or inferring those models are computationally expensive
- Our contribution

Confidence loss for training more plausible simple DNNs	GAN for generating out-of-distribution samples	Joint training method of classifier and GAN

  - Experimental results
    - Our method drastically improves the detection performance
    - E.g., VGGNet trained by our method improves TPR compared to the baseline: 14.0%→39.1% and 46.3% → 98.9% on CIFAR-10 and SVHN

Contribution 1: Confident Loss

- Confident loss
  - Minimize the KL divergence on data from out-of-distribution

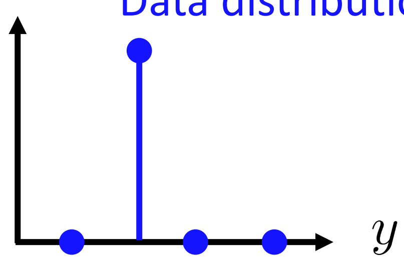
$$\boxed{\min_{\theta}\ \underline{\mathbb{E}_{P_{\mathrm{in}}(\widehat{\mathbf{x}},\widehat{y})}}\Big[-\log P_{\theta}\left(y=\widehat{y}|\widehat{\mathbf{x}}\right)\Big]+\beta\ \underline{\mathbb{E}_{P_{\mathrm{out}}(\mathbf{x})}}\Big[KL\left(\mathcal{U}\left(y\right)\parallel P_{\theta}\left(y|\mathbf{x}\right)\right)\Big],}$$

Data from in-dist Data from out-of-dist

  - Interpretation
    - Assigning higher maximum prediction values to in-distribution samples than o ut-of-distribution ones

$P_{\theta}(y|\mathbf{x}) \rightarrow P(y|\mathbf{x})$

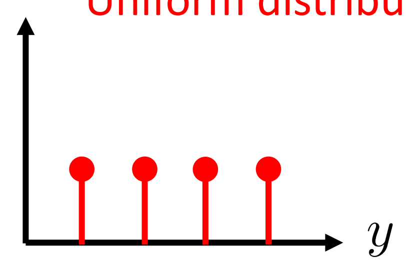
Data distribution



[In-distribution data]

$P_{\theta}(y|\mathbf{x}) \rightarrow \mathcal{U}(y)$

Uniform distribution



[Out-of-distribution data]

"Zero confidence"

Contribution 1: Confident Loss

- Confident loss
  - Minimize the KL divergence on data from out-of-distribution

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

Data from in-dist　　　　Data from out-of-dist

  - Interpretation
    - Assigning higher maximum prediction values to in-distribution samples than out-of-distribution ones
  - Effects of confidence loss
    - Fraction of the maximum prediction value from simple CNNs (2 Conv + 3 FC)

SVHN

CIFAR-10

TinyImageNet

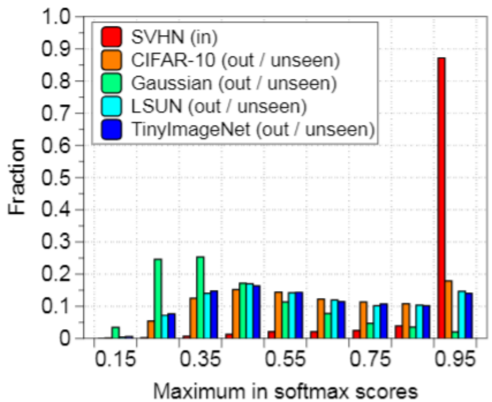
LSUN

Contribution 1: Confident Loss

- Confident loss
  - Minimize the KL divergence on data from out-of-distribution

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

  - Interpretation
    - Assigning higher maximum prediction values to in-distribution samples than o ut-of-distribution ones
  - Effects of confidence loss
    - Fraction of the maximum prediction value from simple CNNs (2 Conv + 3 FC)
    - In-distribution: SVHN



(a) Cross entropy loss

Contribution 1: Confident Loss

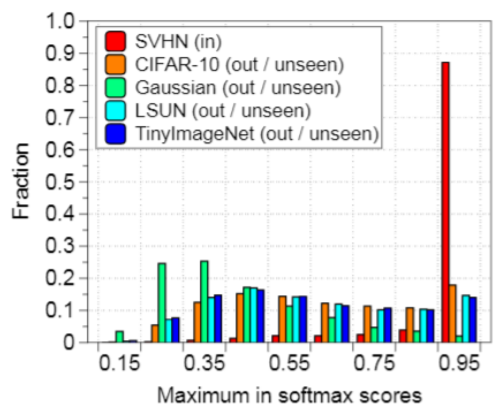
- Confident loss
  - Minimize the KL divergence on data from out-of-distribution

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

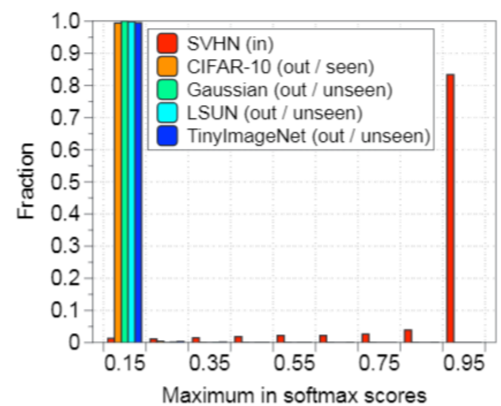
Data from in-dist

Data from out-of-dist

  - Interpretation
    - Assigning higher maximum prediction values to in-distribution samples than o ut-of-distribution ones
  - Effects of confidence loss
    - Fraction of the maximum prediction value from simple CNNs (2 Conv + 3 FC)
    - KL divergence term is optimized using CIFAR-10 training data



(a) Cross entropy loss



(b) Confidence loss in (1)

Contribution 2. GAN for Generating Out-of-Distribution Samples

- Main issues of confidence loss
  - How to optimize the KL divergence loss?

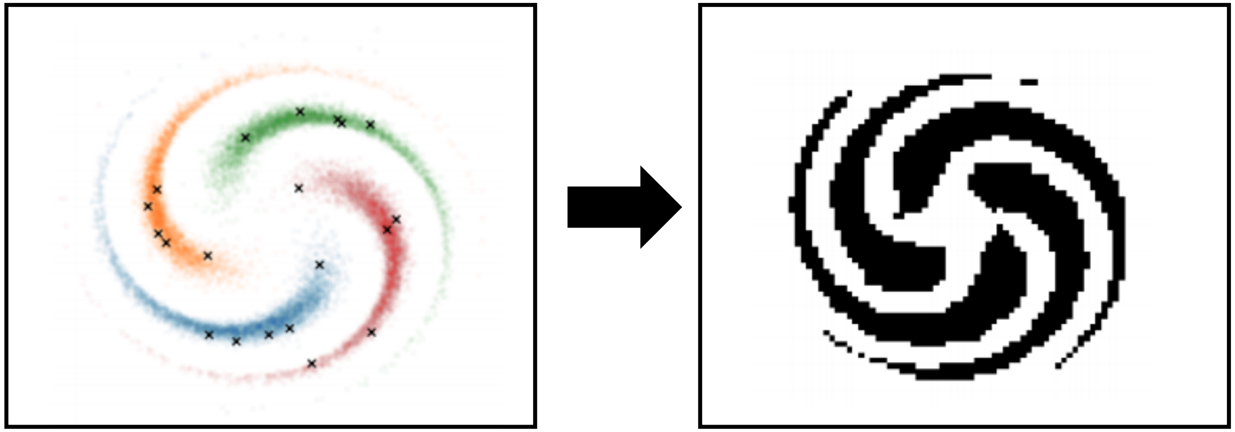
$$\boxed{\min_{\theta} \ \mathbb{E}_{P_{\mathtt{in}}(\widehat{\mathbf{x}},\widehat{y})}\Big[-\log P_{\theta}\left(y=\widehat{y}|\widehat{\mathbf{x}}\right)\Big] + \beta \ \underline{\mathbb{E}_{P_{\mathtt{out}}(\mathbf{x})}}\Big[KL\left(\mathcal{U}\left(y\right) \parallel P_{\theta}\left(y|\mathbf{x}\right)\right)\Big],}$$

Data from out-of-dist

Contribution 2. GAN for Generating Out-of-Distribution Samples

- Main issues of confidence loss
  - How to optimize the KL divergence loss?
    - The number of out-of-distribution samples might be almost infinite to cover the entire space

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



Contribution 2. GAN for Generating Out-of-Distribution Samples

- Main issues of confidence loss
  - How to optimize the KL divergence loss?
    - The number of out-of-distribution samples might be almost infinite to cover the entire space

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

Data from out-of-dist

Contribution 2. GAN for Generating Out-of-Distribution Samples

- Main issues of confidence loss
  - How to optimize the KL divergence loss?
    - The number of out-of-distribution samples might be almost infinite to cover the entire space

- Our intuition
  - Samples close to in-distribution could be more effective in improving the detection performance

Contribution 2. GAN for Generating Out-of-Distribution Samples

- Main issues of confidence loss
  - How to optimize the KL divergence loss?
    - The number of out-of-distribution samples might be almost infinite to cover the entire space

- Our intuition
  - Samples close to in-distribution could be more effective in improving the detection performance

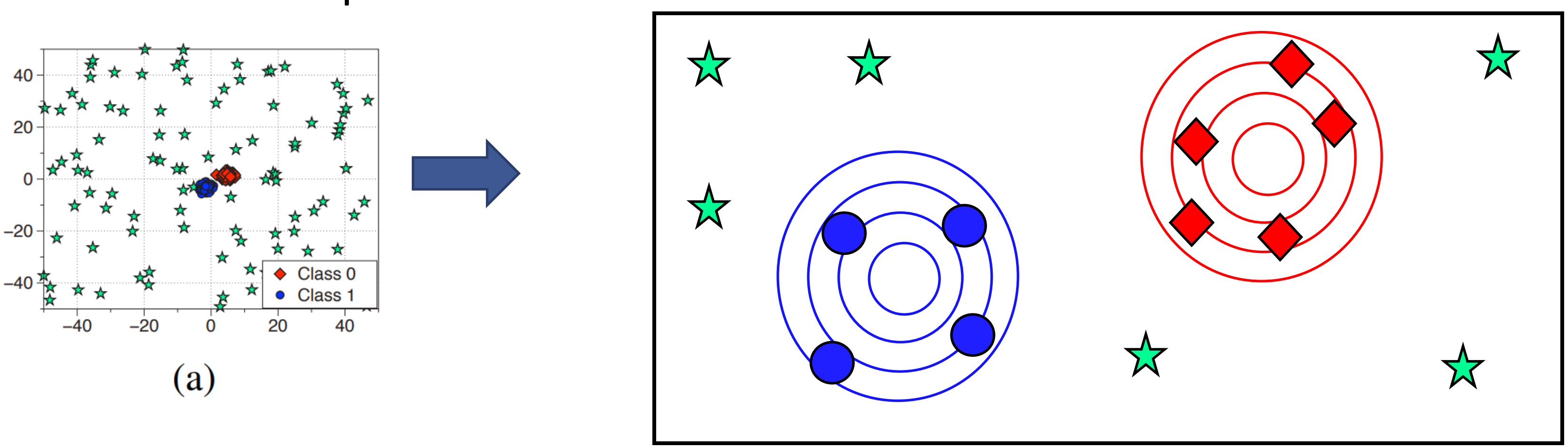
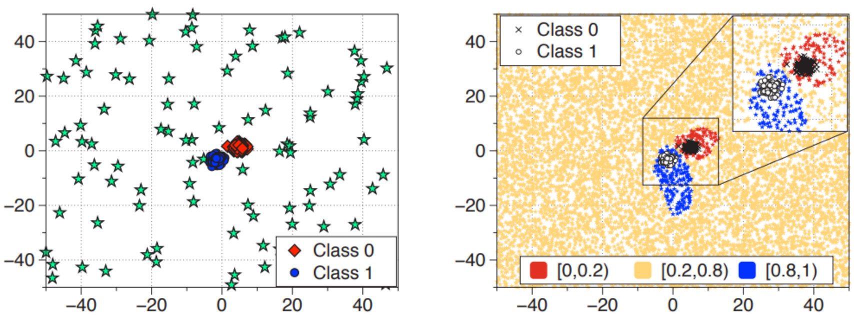


Figure 2: Illustrating the behavior of classifier under different datasets. We generate the out-of-distribution samples from (a) 2D box $[-50, 50]^2$, and show (b) the corresponding decision boundary of classifier. We also generate the out-of-distribution samples from (c) 2D box $[-20, 20]^2$, and show (d) the corresponding decision boundary of classifier.

Contribution 2. GAN for Generating Out-of-Distribution Samples

- Main issues of confidence loss
  - How to optimize the KL divergence loss?
    - The number of out-of-distribution samples might be almost infinite to cover the entire space
- Our intuition
  - Samples close to in-distribution could be more effective in improving the detection performance



(a) (b)

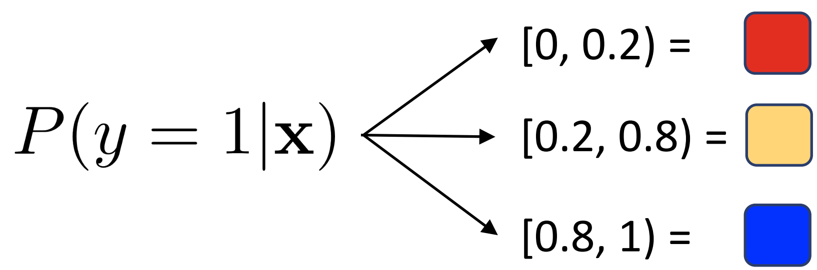


Figure 2: Illustrating the behavior of classifier under different datasets. We generate the out-of-distribution samples from (a) 2D box $[-50, 50]^2$, and show (b) the corresponding decision boundary of classifier. We also generate the out-of-distribution samples from (c) 2D box $[-20, 20]^2$, and show (d) the corresponding decision boundary of classifier.

Contribution 2. GAN for Generating Out-of-Distribution Samples

- Main issues of confidence loss
  - How to optimize the KL divergence loss?
    - The number of out-of-distribution samples might be almost infinite to cover the entire space

- Our intuition
  - Samples close to in-distribution could be more effective in improving the detection performance

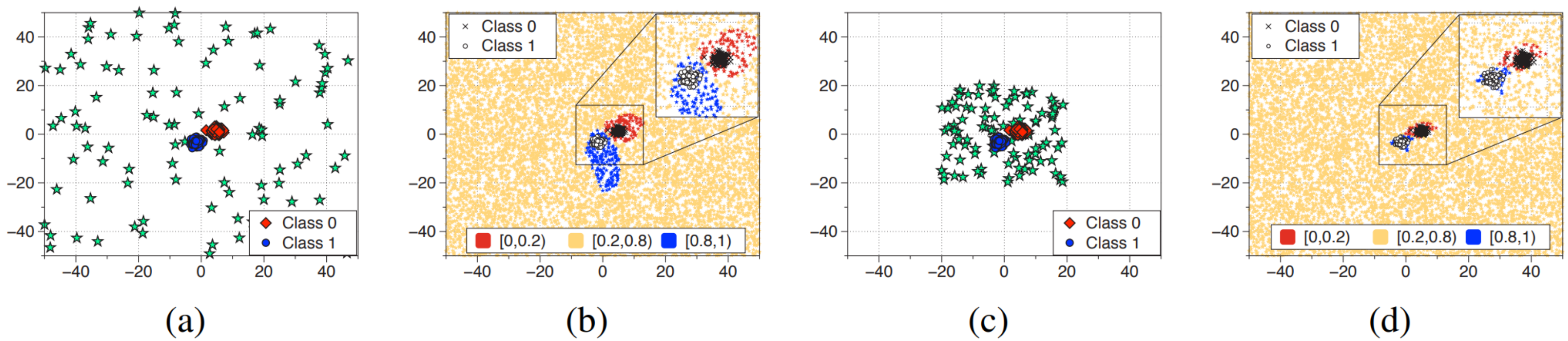


Figure 2: Illustrating the behavior of classifier under different datasets. We generate the out-of-distribution samples from (a) 2D box $[-50, 50]^2$, and show (b) the corresponding decision boundary of classifier. We also generate the out-of-distribution samples from (c) 2D box $[-20, 20]^2$, and show (d) the corresponding decision boundary of classifier.

Contribution 2. GAN for Generating Out-of-Distribution Samples

- New GAN objective

$$\min_G \max_D$$

$$+ \underbrace{\mathbb{E}_{P_{\text{in}}(\mathbf{x})}\big[\log D(\mathbf{x})\big] + \mathbb{E}_{P_G(\mathbf{x})}\big[\log(1 - D(\mathbf{x}))\big]}_{\text{(b)}},$$

  - Term (b) corresponds to the original GAN loss
    - Generating out-of-distribution samples close to in-distribution

Contribution 2. GAN for Generating Out-of-Distribution Samples

- New GAN objective

$$\boxed{\begin{aligned}\min_G \max_D \quad & \beta\, \underbrace{\mathbb{E}_{P_G(\mathbf{x})}\Big[KL\left(\mathcal{U}\left(y\right) \parallel P_\theta\left(y|\mathbf{x}\right)\right)\Big]}_{\text{(a)}} \\ & + \underbrace{\mathbb{E}_{P_{\text{in}}(\mathbf{x})}\Big[\log D\left(\mathbf{x}\right)\Big] + \mathbb{E}_{P_G(\mathbf{x})}\Big[\log\left(1 - D\left(\mathbf{x}\right)\right)\Big]}_{\text{(b)}},\end{aligned}}$$

  - Term (a) forces the generator to generate low-density samples
    - (approximately) minimizing the log negative likelihood of in-distribution
  - Term (b) corresponds to the original GAN loss
    - Generating out-of-distribution samples close to in-distribution

$$P_{\text{in}}(\mathbf{x}) \approx \exp\left(KL\left(\mathcal{U}\left(y\right) \parallel P_\theta\left(y|\mathbf{x}\right)\right)\right)$$

Contribution 2. GAN for Generating Out-of-Distribution Samples

- New GAN objective

Classifier trained on in-distribution

$$\min_G \max_D \quad \underbrace{\beta\, \mathbb{E}_{P_G(\mathbf{x})}\left[KL\left(\mathcal{U}\left(y\right) \parallel \boxed{P_\theta\left(y|\mathbf{x}\right)}\right)\right]}_{(a)}$$

$$+ \underbrace{\mathbb{E}_{P_{\text{in}}(\mathbf{x})}\left[\log D\left(\mathbf{x}\right)\right] + \mathbb{E}_{P_G(\mathbf{x})}\left[\log\left(1 - D\left(\mathbf{x}\right)\right)\right]}_{(b)},$$

$$P_{\text{in}}(\mathbf{x}) \approx \exp\left(KL\left(\mathcal{U}\left(y\right) \parallel P_\theta\left(y|\mathbf{x}\right)\right)\right)$$

  - Term (a) forces the generator to generate low-density samples
    - (approximately) minimizing the log negative likelihood of in-distribution
  - Term (b) corresponds to the original GAN loss
    - Generating out-of-distribution samples close to in-distribution

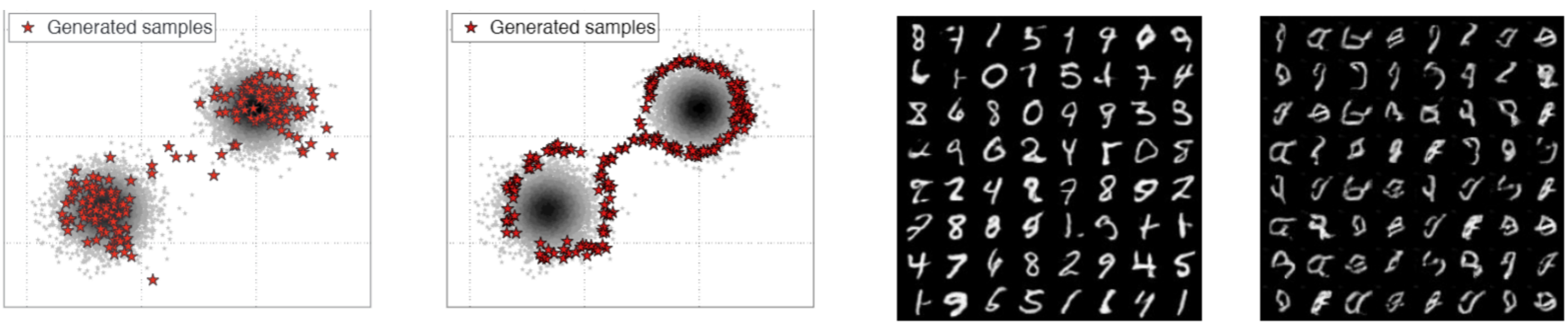
Contribution 2. GAN for Generating Out-of-Distribution Samples

- New GAN objective

$$\min_G \max_D \quad \beta \underbrace{\mathbb{E}_{P_G(\mathbf{x})}\Big[KL\left(\mathcal{U}\left(y\right) \parallel P_\theta\left(y|\mathbf{x}\right)\right)\Big]}_{\text{(a)}} + \underbrace{\mathbb{E}_{P_{\text{in}}(\mathbf{x})}\Big[\log D\left(\mathbf{x}\right)\Big] + \mathbb{E}_{P_G(\mathbf{x})}\Big[\log\left(1 - D\left(\mathbf{x}\right)\right)\Big]}_{\text{(b)}},$$

  - Term (a) forces the generator to generate low-density samples
    - (approximately) minimizing the log negative likelihood of in-distribution
  - Term (b) corresponds to the original GAN loss
    - Generating out-of-distribution samples close to in-distribution

- Experimental results on toy example and MNIST



(a) (b) (c) (d)

Figure 3: The generated samples from original GAN (a)/(c) and proposed GAN (b)/(d).

Contribution 3. Joint Confidence Loss

- We suggest training the proposed GAN using a confident classifier
    - Converse is also possible

Contribution 3. Joint Confidence Loss

- We suggest training the proposed GAN using a confident classifier
    - Converse is also possible
- We propose a joint confidence loss

$$\boxed{\begin{aligned}\min_G \max_D \min_\theta \quad & \underbrace{\mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})}\Big[-\log P_\theta\left(y=\widehat{y}|\widehat{\mathbf{x}}\right)\Big]}_{\text{(c)}} + \beta\,\underbrace{\mathbb{E}_{P_G(\mathbf{x})}\Big[KL\left(\mathcal{U}\left(y\right)\parallel P_\theta\left(y|\mathbf{x}\right)\right)\Big]}_{\text{(d)}} \\ & \underbrace{+\mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}})}\Big[\log D\left(\widehat{\mathbf{x}}\right)\Big] + \mathbb{E}_{P_G(\mathbf{x})}\Big[\log\left(1-D\left(\mathbf{x}\right)\right)\Big]}_{\text{(e)}}.\end{aligned}}$$

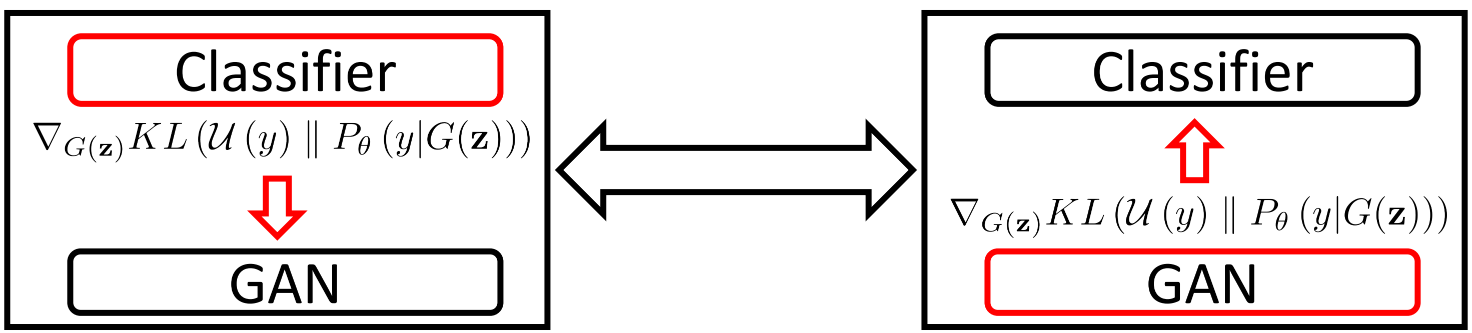
    - Classifier's confidence loss: (c) + (d)
    - GAN loss: (d) + (e)

Contribution 3. Joint Confidence Loss

- We suggest training the proposed GAN using a confident classifier
  - Converse is also possible
- We propose a joint confidence loss

$$\min_{G} \max_{D} \min_{\theta} \quad \underbrace{\mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})}\Big[-\log P_\theta\left(y=\widehat{y}|\widehat{\mathbf{x}}\right)\Big]}_{\text{(c)}} + \beta\, \underbrace{\mathbb{E}_{P_G(\mathbf{x})}\Big[KL\left(\mathcal{U}\left(y\right) \parallel P_\theta\left(y|\mathbf{x}\right)\right)\Big]}_{\text{(d)}}$$
$$\underbrace{+\mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}})}\Big[\log D\left(\widehat{\mathbf{x}}\right)\Big] + \mathbb{E}_{P_G(\mathbf{x})}\Big[\log\left(1-D\left(\mathbf{x}\right)\right)\Big].}_{\text{(e)}}$$

  - Classifier's confidence loss: (c) + (d)
  - GAN loss: (d) + (e)
- Alternating algorithm for optimizing the joint confidence loss



Step 1. update GAN

Step 2. update classifier

Experimental Results: dataset & model

- Model: VGGNet [Christian’ 15] with 13 layers
- In-distribution: CIFAR-10 or SVHN

CIFAR-10 [Krizhevsky’ 09]

- 32×32 RGB
- 10 classes
- 50,000 training set
- 10,000 test set

SVHN [Netzer’ 11]

- 32×32 RGB
- 10 classes
- 73,257 training set
- 26,032 test set

- Out-of-distribution: (resized) TinyImageNet and LSUN

TinyImageNet

- 32×32 RGB
- 200 classes
- 10,000 test set

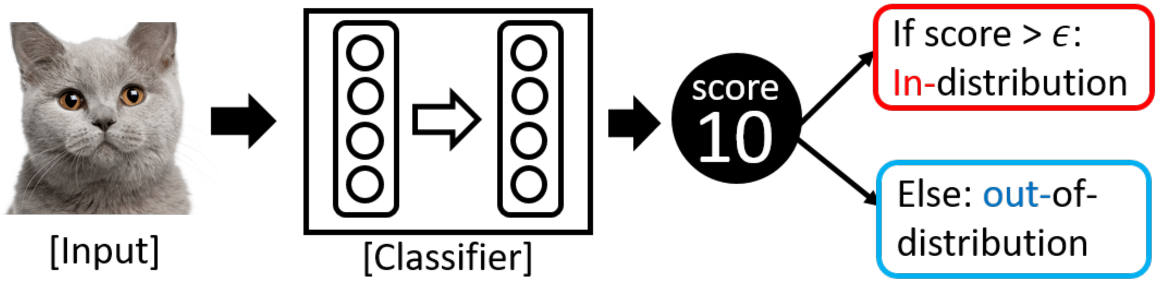
LSUN

- 32×32 RGB
- 10 classes
- 10,000 test set

[Krizhevsky’ 09] Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Master’s thesis, Department of Computer Science, University of Toronto, 2009.
[Netzer’ 11] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. and Ng, A.Y. Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.
[Christian’ 15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Computer Vision and Pattern Recognition (CVPR), 2015

Experimental Results - Metric

- TP = true positive
- FN = false negative
- TN = true negative
- FP = false positive



- **[Metrics]**
- **FPR at 95% TPR**
  - FPR = FP/(FP + TN), TPR = TP/(TP + FN)
- **AUROC (Area Under the Receiver Operating Characteristic curve)**
  - ROC curve = relationship between TPR and FPR
- **Detection Error**
  - Minimum misclassification probability over all thresholds

$$\boxed{\min_{\delta} \left\{ H\left(g\left(\mathbf{x};\sigma\right) \neq 1 | z=1\right) H\left(z=1\right) + H\left(g\left(\mathbf{x};\sigma\right) \neq 0 | z=0\right) H\left(z=0\right) \right\}}$$

- **AUPR (Area under the Precision-Recall curve)**
  - PR curve = relationship between precision=TP/(TP+FP) and recall=TP/(TP+FN)

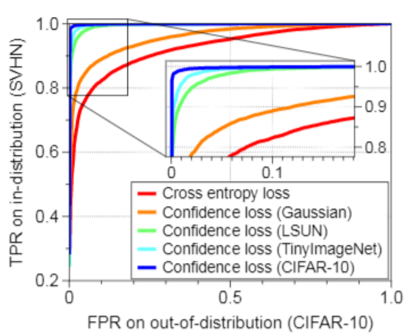
Experimental Results

- Measure the detection performance of threshold-based detectors
- Confidence loss with some explicit out-of-distribution dataset

In-dist	Out-of-dist	Classification accuracy	TNR at TPR 95%	AUROC	Detection accuracy	AUPR in	AUPR out
		Cross entropy loss / Confidence loss					
SVHN	CIFAR-10 (seen)	93.82 / **94.23**	47.4 / **99.9**	62.6 / **99.9**	78.6 / **99.9**	71.6 / **99.9**	91.2 / **99.4**
	TinyImageNet (unseen)		49.0 / **100.0**	64.6 / **100.0**	79.6 / **100.0**	72.7 / **100.0**	91.6 / **99.4**
	LSUN (unseen)		46.3 / **100.0**	61.8 / **100.0**	78.2 / **100.0**	71.1 / **100.0**	90.8 / **99.4**
	Gaussian (unseen)		56.1 / **100.0**	72.0 / **100.0**	83.4 / **100.0**	77.2 / **100.0**	92.8 / **99.4**
CIFAR-10	SVHN (seen)	80.14 / **80.56**	13.7 / **99.8**	46.6 / **99.9**	66.6 / **99.8**	61.4 / **99.9**	73.5 / **99.8**
	TinyImageNet (unseen)		**13.6** / 9.9	**39.6** / 31.8	**62.6** / 58.6	**58.3** / 55.3	**71.0** / 66.1
	LSUN (unseen)		**14.0** / 10.5	**40.7** / 34.8	**63.2** / 60.2	**58.7** / 56.4	**71.5** / 68.0
	Gaussian (unseen)		2.8 / **3.3**	10.2 / **14.1**	50.0 / 50.0	48.1 / **49.4**	39.9 / **47.0**

Table 1: Performance of the baseline detector (Hendrycks & Gimpel, 2016) using VGGNet. All values are percentages and boldface values indicate relative the better results. For each in-distribution, we minimize the KL divergence term in (1) using training samples from an out-of-distribution dataset denoted by "seen", where other "unseen" out-of-distributions were only used for testing.

- Classifier trained by our method drastically improves the detection performance across all out-of-distributions

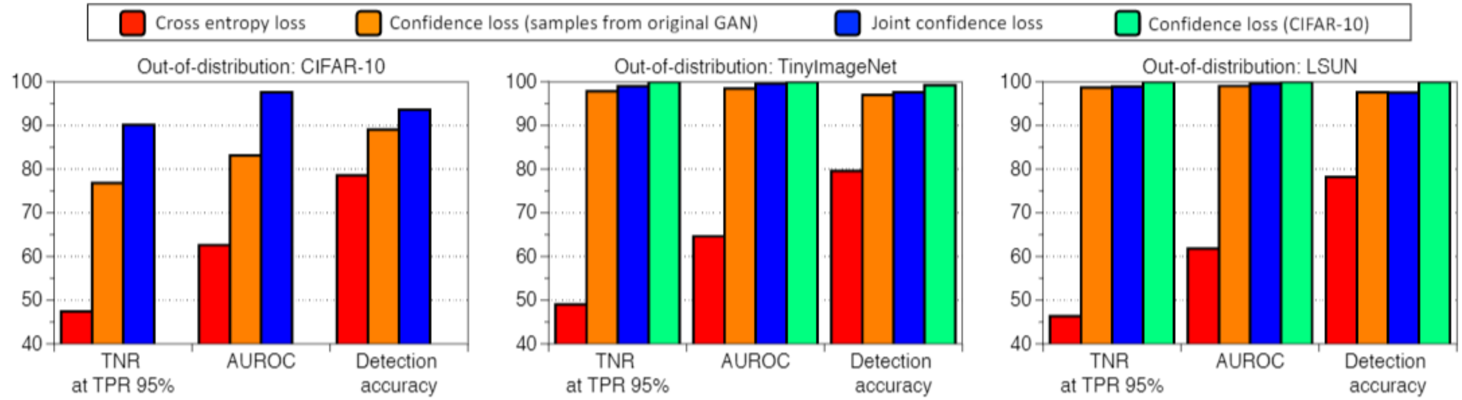


(c) ROC curve

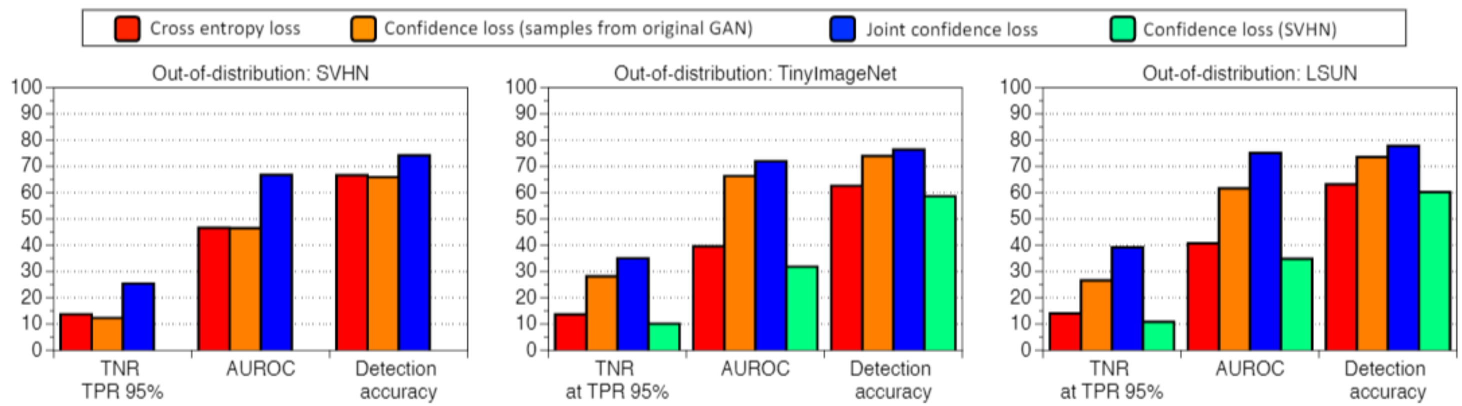
⇨ Realistic images such as TinyImageNet (aqua line) and LSUN(green line) are more useful than synthetic datasets (orange line) for improving the detection perfor-mance

Experimental Results

- Joint confidence loss



(a) In-distribution: SVHN

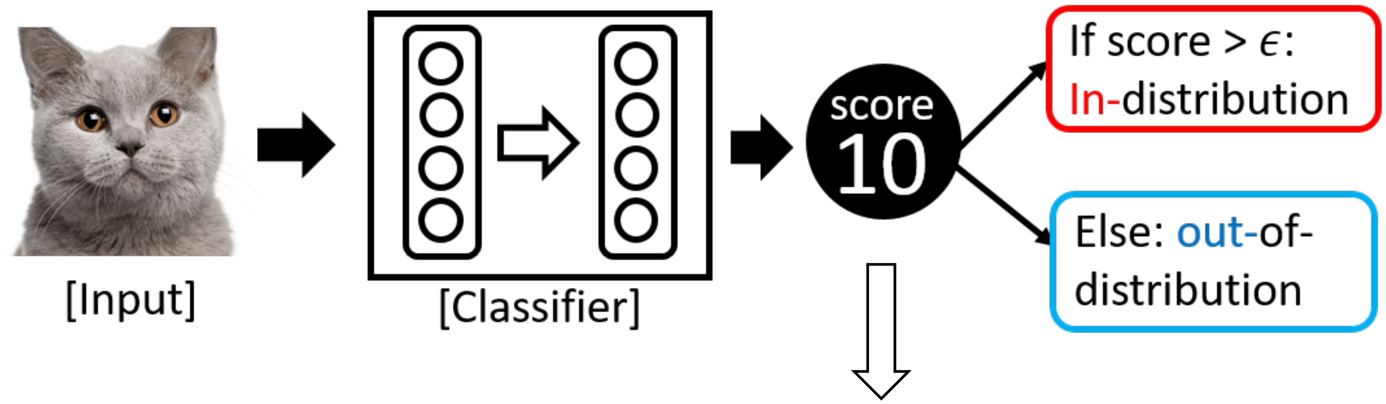


(b) In-distribution: CIFAR-10

- Confidence loss with the original GAN (orange bar) is often useful for improving the detection performance
- Joint confidence loss (bluebar) still outperforms all baseline it in all cases

Experimental Results

- Comparison with ODIN [Liang’ 18]



$$S_i(\boldsymbol{x};T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^{N}\exp\left(f_j(\boldsymbol{x})/T\right)}, \quad + \quad \tilde{\boldsymbol{x}} = \boldsymbol{x} - \varepsilon \text{sign}(-\nabla_{\boldsymbol{x}} \log S_{\hat{y}}(\boldsymbol{x};T)),$$

$$g(\boldsymbol{x};\delta,T,\varepsilon) = \begin{cases} 1 & \text{if} \max_i p(\tilde{\boldsymbol{x}};T) \le \delta, \\ 0 & \text{if} \max_i p(\tilde{\boldsymbol{x}};T) > \delta. \end{cases}$$

Experimental Results

- Comparison with ODIN [Liang' 18]

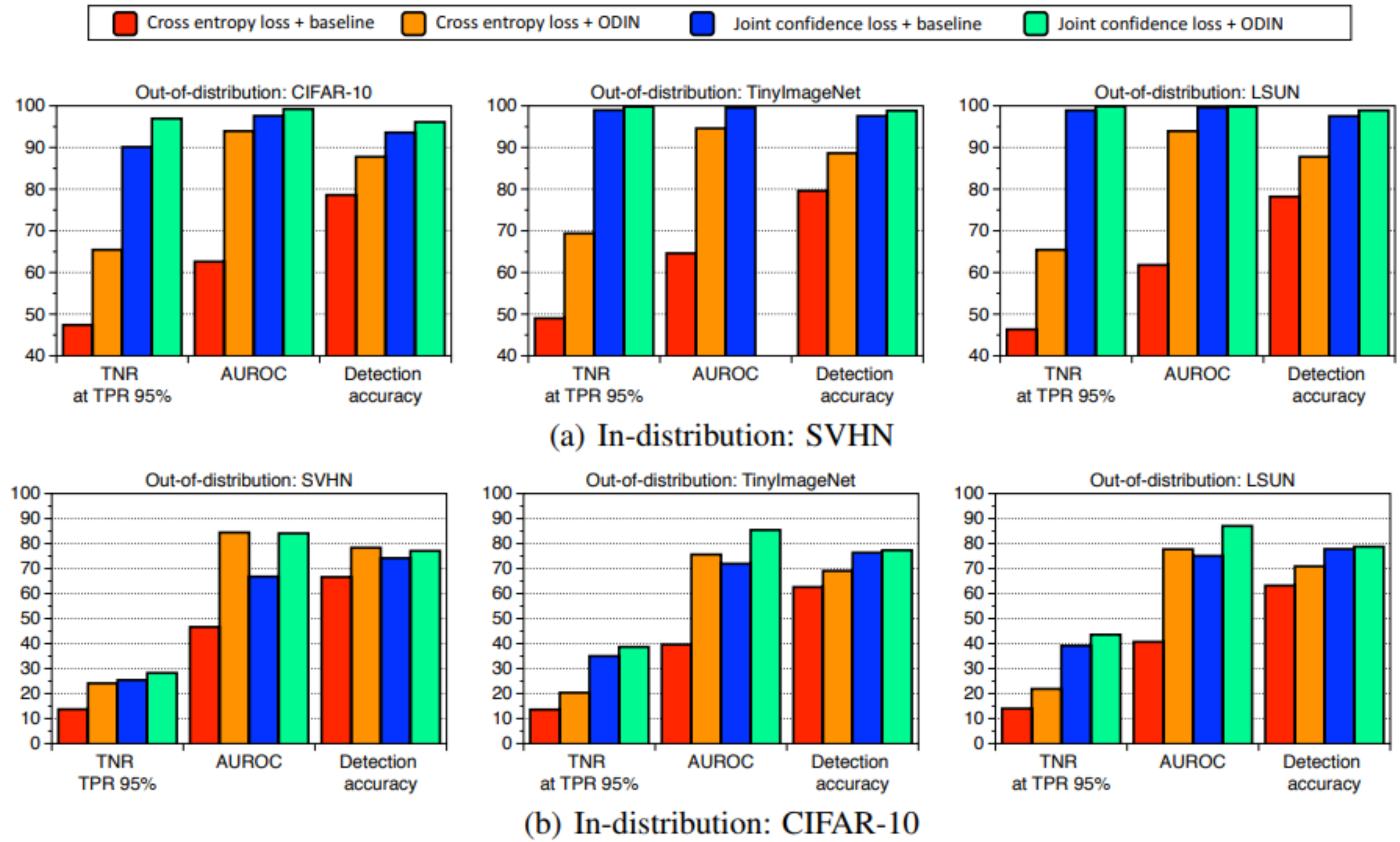
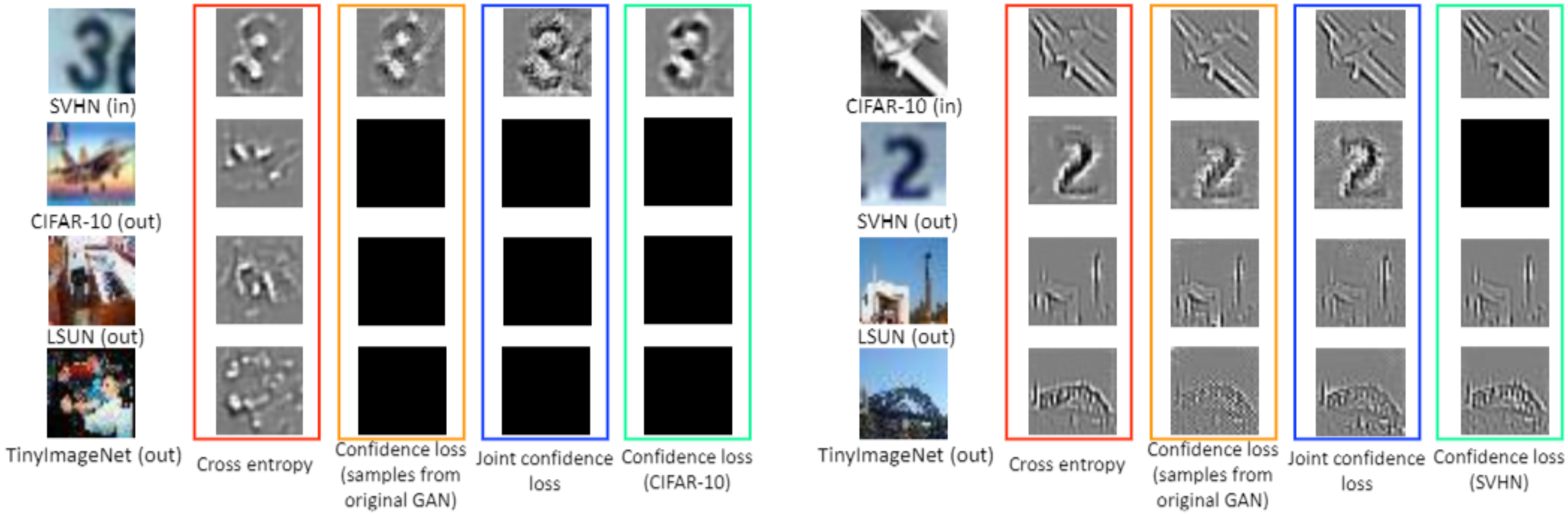


Figure 7: Performances of the baseline detector (Hendrycks & Gimpel, 2016) and ODIN detector (Liang et al., 2017) under various training losses.

Experimental Results

- Interpretability of trained classifier



(a) In-distribution: SVHN (b) In-distribution: CIFAR-10

Figure 5: Guided gradient (sensitivity) maps of the top-1 predicted class with respect to the input image under various training losses.

- Classifier trained by cross entropy loss shows sharp gradient maps for both samples from in- and out-of-distributions
- Classifiers trained by the confidence losses do only on samples from in-distribution.

Outline

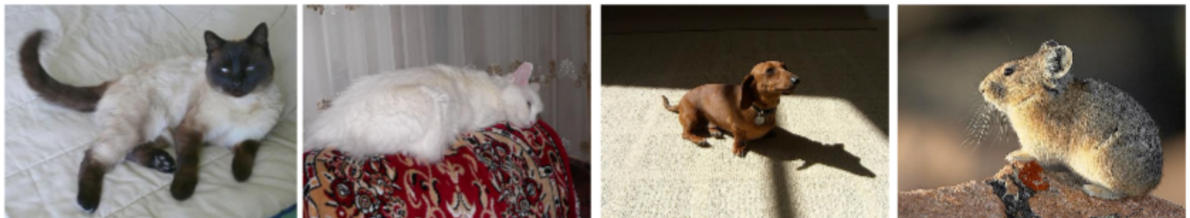
- Introduction
  - Predictive uncertainty of deep neural networks
  - Summary of contributions
- How to train confident neural networks
  - Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples [Lee' 18a]
- Applications
  - Hierarchical novelty detection [Lee' 18b]
- Conclusion
  - Future work

[Lee' 18a] Lee, K., Lee, H., Lee, K. and Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. In ICLR, 2018.
[Lee' 17] Lee, K., Hwang, C., Park, K. and Shin, J. Confident Multiple Choice Learning. *In ICML, 2017.*
[Lee' 18b] Lee, K., Lee, Min. K, Zhang, Y. Shin. J, Lee, H. Hierarchical Novelty Detection for Visual Object Recognition, In CVPR, 2018.

Hierarchical Novelty Detection

- Novelty detection

Test image:



True label:	Siamese cat	Angola cat	Dachshund	Pika
Prior works:	Siamese cat	novel	novel	novel

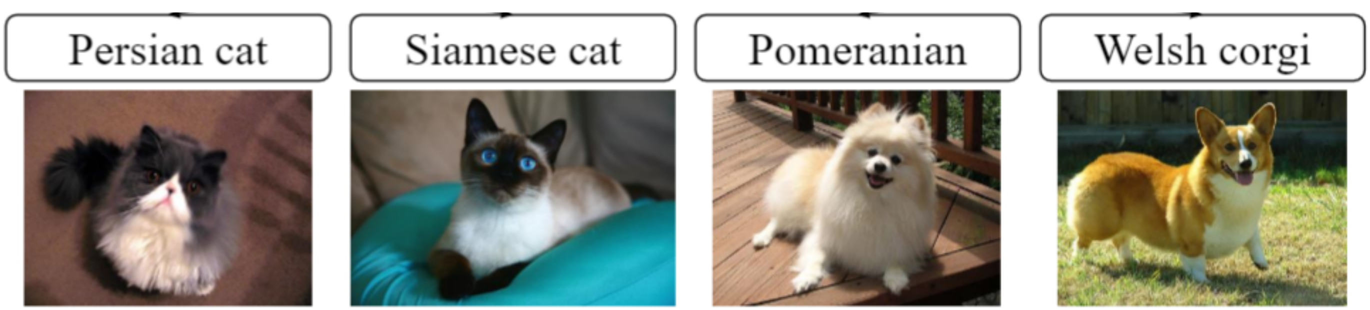


Figure 1. An illustration of our hierarchical novelty detection task

Hierarchical Novelty Detection

- Objective

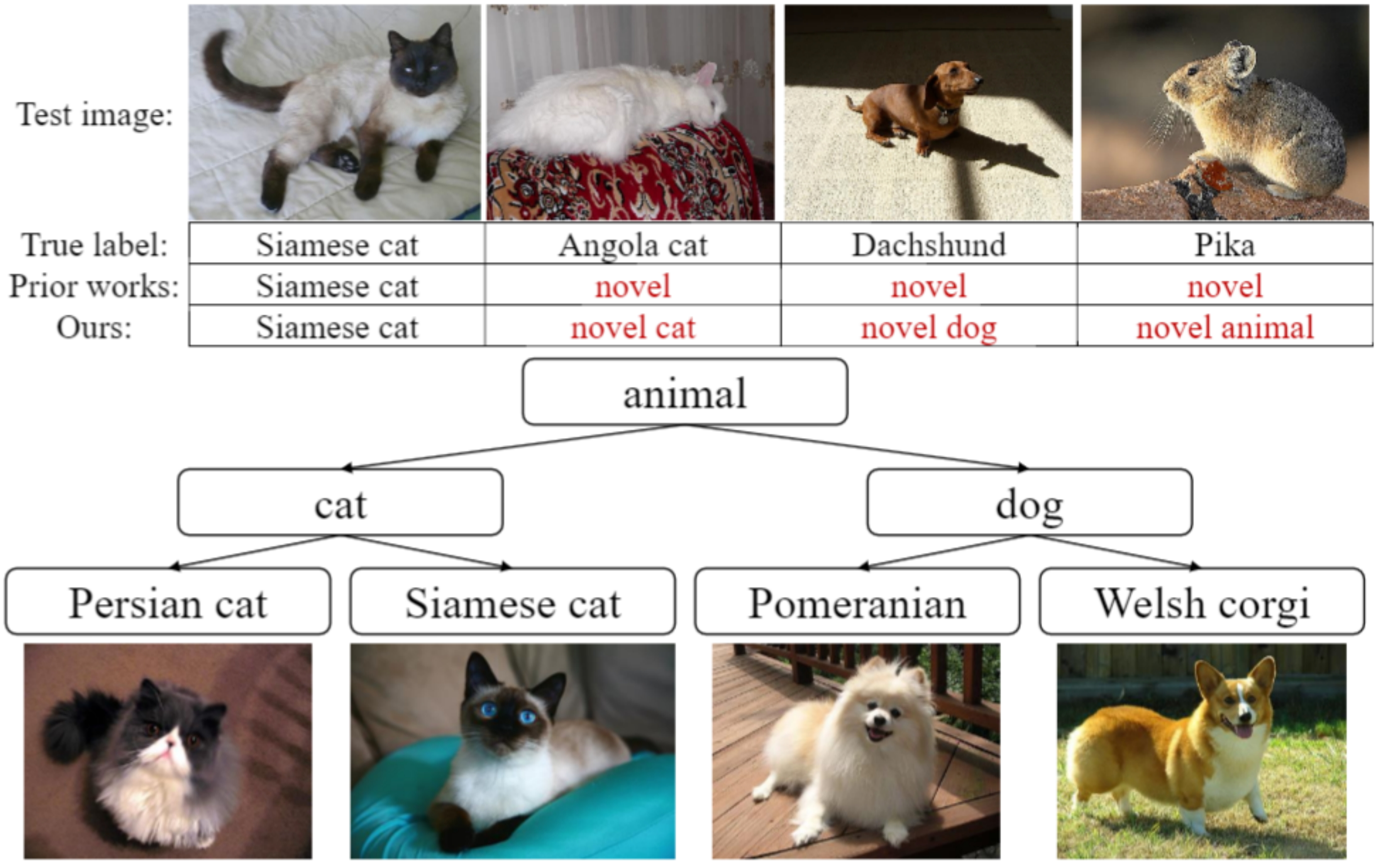


Figure 1. An illustration of our hierarchical novelty detection task

Hierarchical Novelty Detection

- Objective
  - 1. Find the closest known (super-)category in taxonomy
  - 2. Find fine-grained classification for novel categories (i.e., out-of-distribution samples)

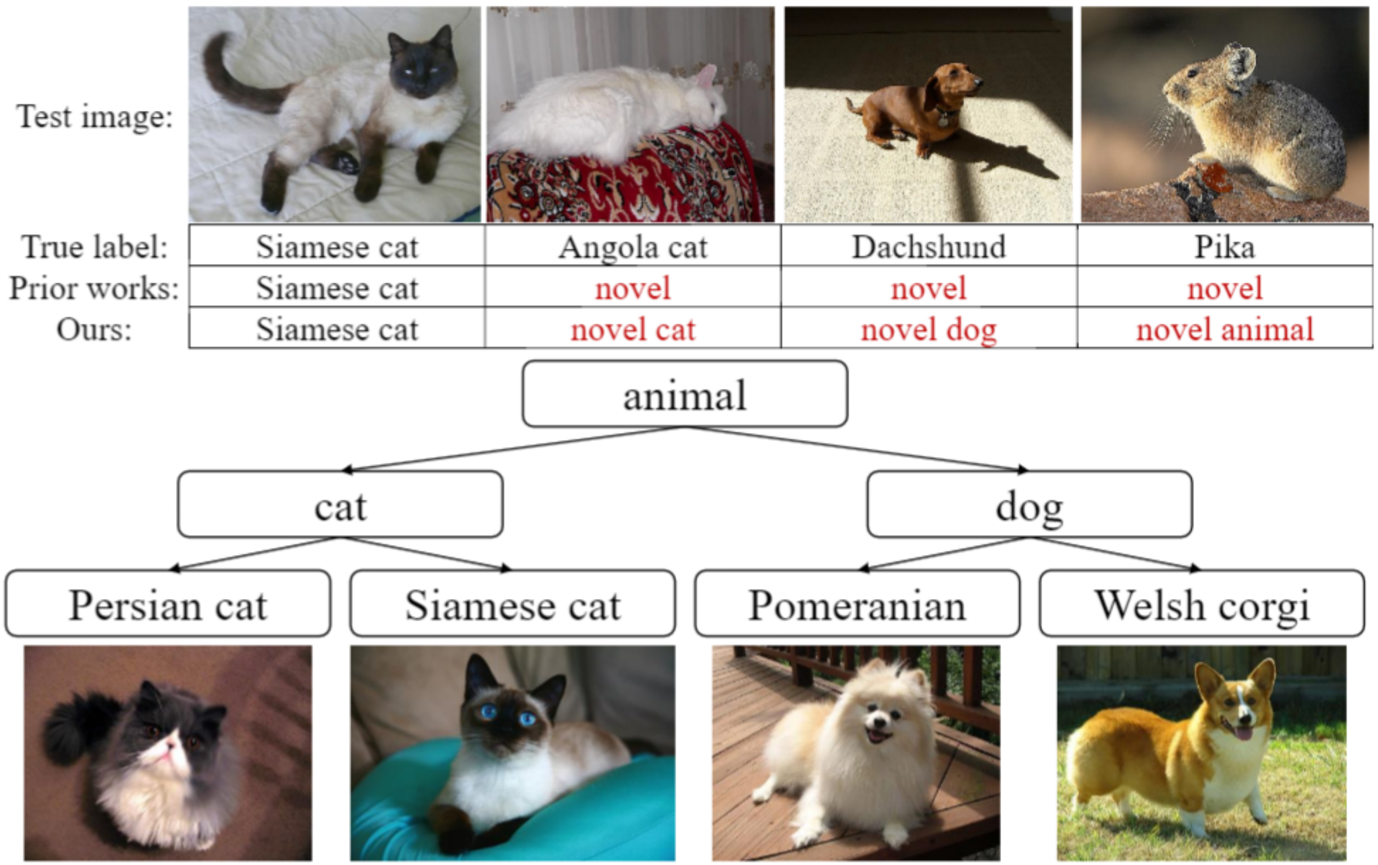
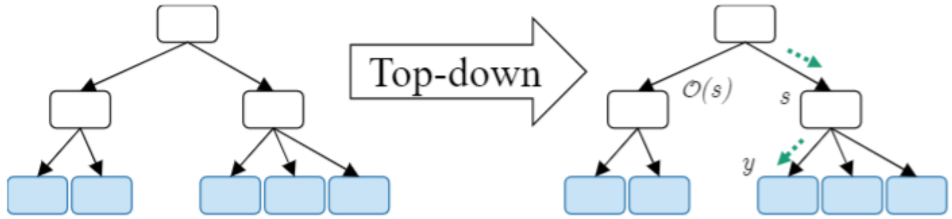


Figure 1. An illustration of our hierarchical novelty detection task

Two Main Approaches

- Top-down method (TD)
  - p(child) = $\sum_{super}$ p(child | super) p(super)



  - Inference

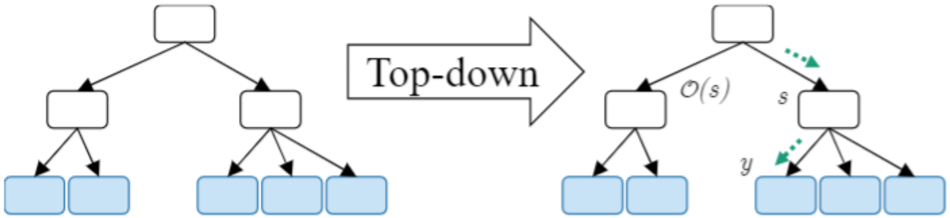
$$\hat{y} = \begin{cases} \arg\max_{y'} \; Pr(y'|x, s; \theta_s) & \text{if confident,} \\ \mathcal{N}(s) & \text{otherwise,} \end{cases}$$

Novel class

- Definition of confidence: $D_{KL}(U(y|s) \parallel Pr(y|x, s; \theta_s)) \geq \lambda_s,$

Two Main Approaches

- Top-down method (TD)
  - p(child) = $\sum_{super}$ p(child | super) p(super)



  - Inference

$$\hat{y} = \begin{cases} \arg\max_{y'} \; Pr(y'|x, s; \theta_s) & \text{if confident,} \\ \mathcal{N}(s) & \text{otherwise,} \end{cases}$$

Novel class

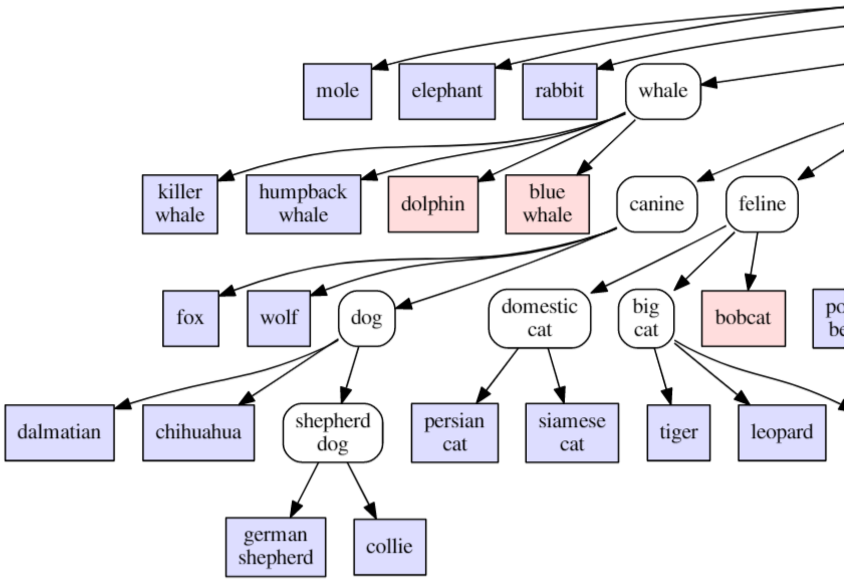
    - Definition of confidence: $D_{KL}(U(y|s) \parallel Pr(y|x, s; \theta_s)) \geq \lambda_s,$
  - Objective

$$\boxed{\begin{aligned} \min_{\theta_s} \; & \mathbb{E}_{Pr(x,y|s)} \left[-\log Pr(y|x, s; \theta_s) \right] \\ & + \mathbb{E}_{Pr(x,y|\mathcal{O}(s))} \left[D_{KL} \left(U(y|s) \parallel Pr(y|x, s; \theta_s) \right) \right], \end{aligned}}$$

$Pr(x, y|\mathcal{O}(s))$ denotes the data distribution of all exclusive classes from s

Experimental Results on ImageNet Dataset

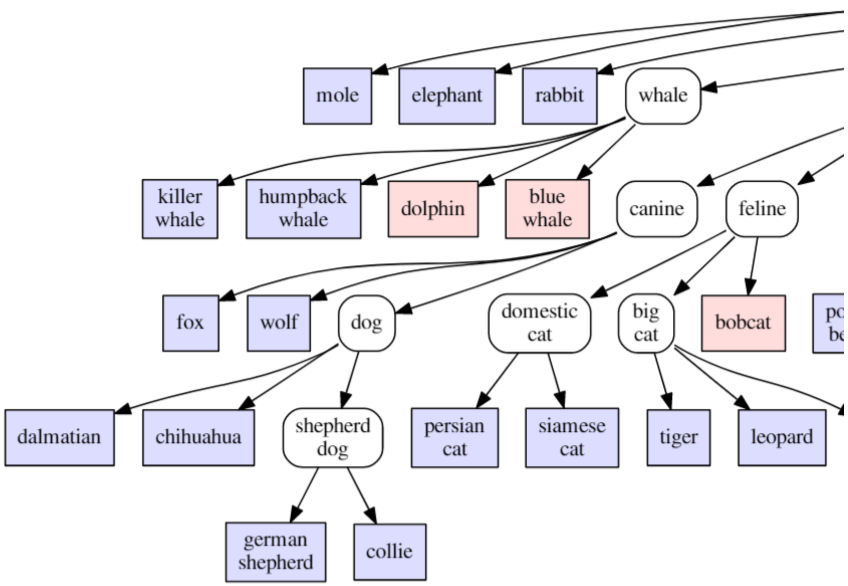
- ImageNet dataset
  - 22K classes
  - Taxonomy
    - 396 super classes of 1K known leaf classes
    - Rest of 21K classes can be used as novel class
  - Example



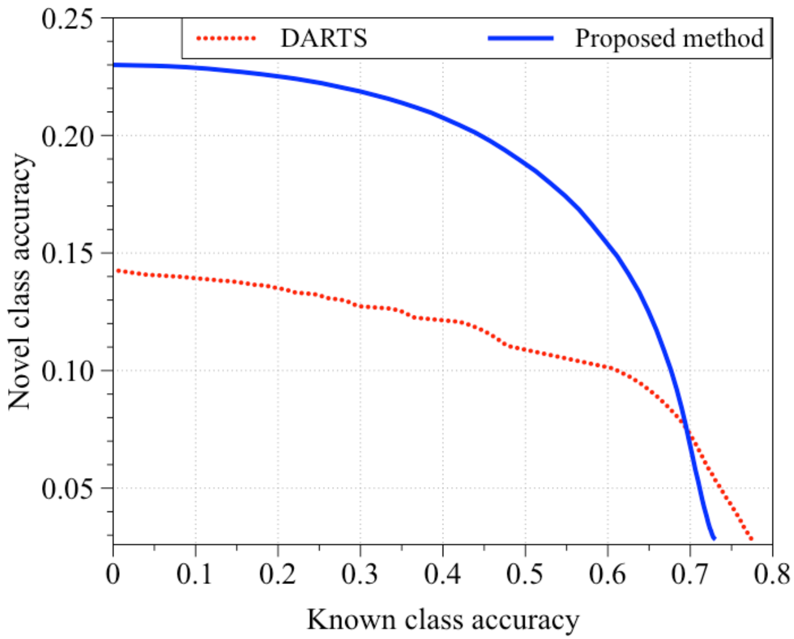
[Deng' 12] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade offs in large scale visual recognition. In CVPR , pages 3450–3457. IEEE, 2012.

Experimental Results on ImageNet Dataset

- ImageNet dataset
  - 22K classes
  - Taxonomy
    - 396 super classes of 1K known leaf classes
    - Rest of 21K classes can be used as novel class
  - Example



- Hierarchical novelty detection performance
  - Baseline: DARTS [Deng' 12]



- One can note that our methods have higher novel class accuracy than DARTS to have a same known class accuracy in most regions

[Deng' 12] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade offs in large scale visual recognition. In CVPR , pages 3450–3457. IEEE, 2012.

Conclusion

- We propose a new method for training confident deep neural networks
  - It produce the uniform distribution when the input is not from target distribution

- We show that it can be applied to many machine learning problems:
  - Detecting out-of-distribution problem [Lee' 18a]
  - Ensemble learning using deep neural networks [Lee' 17]
  - Hierarchical novelty detection [Lee' 18b]

- We believe that our new approach brings a refreshing angle for developing confident deep networks in many related applications:
  - Network calibration
  - Adversarial example detection
  - Bayesian probabilistic models
  - Semi-supervised learning

[Lee' 18a] Lee, K., Lee, H., Lee, K. and Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. I n ICLR, 2018.

[Lee' 17] Lee, K., Hwang, C., Park, K. and Shin, J. Confident Multiple Choice Learning. *In ICML, 2017.*

[Lee' 18b] Lee, K., Lee, Min. K, Zhang, Y. Shin. J, Lee, H. Hierarchical Novelty Detection for Visual Object Recognition, In CVPR, 2018.