Google AI
Brain Team

Practical Tutorial on Uncertainty and Out-of-distribution Robustness in Deep Learning

Balaji Lakshminarayanan
balajiln@

Google Research, Brain Team

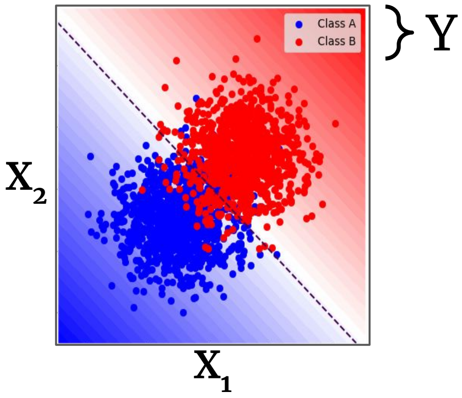
Motivation

What do we mean by Uncertainty?

Return a distribution over predictions rather than a single prediction.

- ***Classification***: Output label along with its confidence.
- ***Regression***: Output mean along with its variance.

Good uncertainty estimates quantify ***when we can trust the model's predictions***.



$$p(\mathbf{y}|\mathbf{x})$$

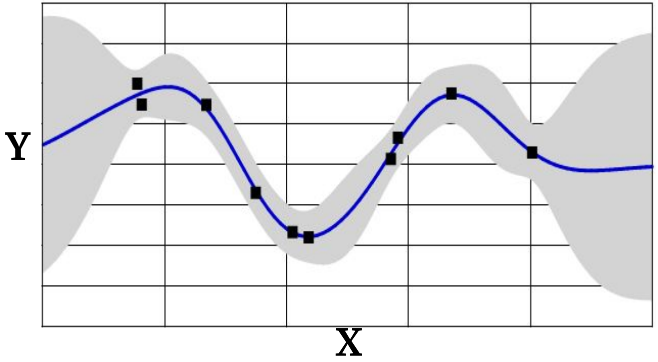


Image credit: Eric Nalisnick

What do we mean by Out-of-Distribution Robustness?

Google AI
Brain Team

I.I.D.

(Independent and Identically Distributed)

$$p_{TEST}(y,x) = p_{TRAIN}(y,x)$$

O.O.D.

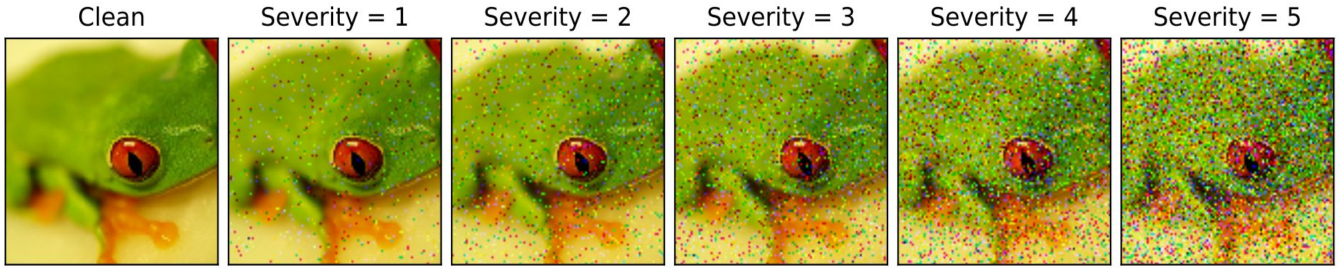
$$p_{TEST}(y,x) \neq p_{TRAIN}(y,x)$$

Examples of dataset shift:

- ***Covariate shift.*** Distribution of features p(x) changes and p(y|x) is fixed.
- ***Open-set recognition.*** New classes may appear at test time.
- ***Subpopulation shift.*** Frequencies of data subpopulations changes.
- ***Label shift.*** Distribution of labels p(y) changes and p(x|y) is fixed.

ImageNet-C: Varying Intensity for Dataset Shift

Google AI Brain Team



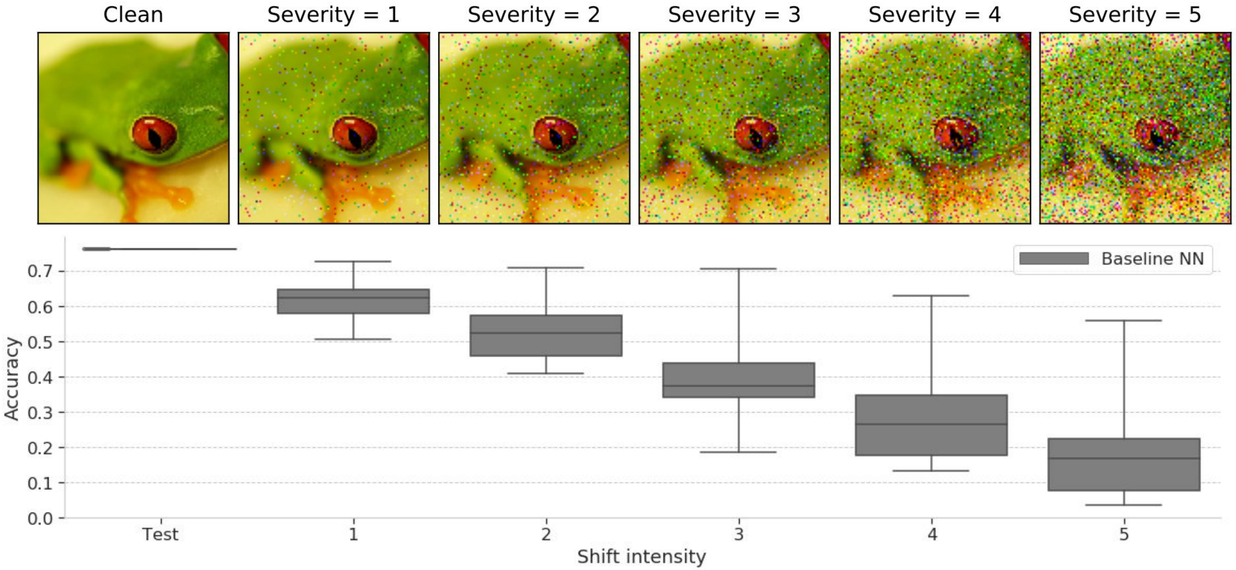
Increasing "OODness"

I.I.D test set

"Distance" between $p_{TEST}(y,x)$ and $p_{TRAIN}(y,x)$ increases

Image source: Benchmarking Neural Network Robustness to Common Corruptions and Perturbations, Hendrycks & Dietterich, 2019.

Neural networks do not generalize under covariate shift



- **Accuracy drops** with increasing shift on Imagenet-C

- But do the models know that they are less accurate?

Neural networks *do not know when they don't know*

Google AI
Brain Team

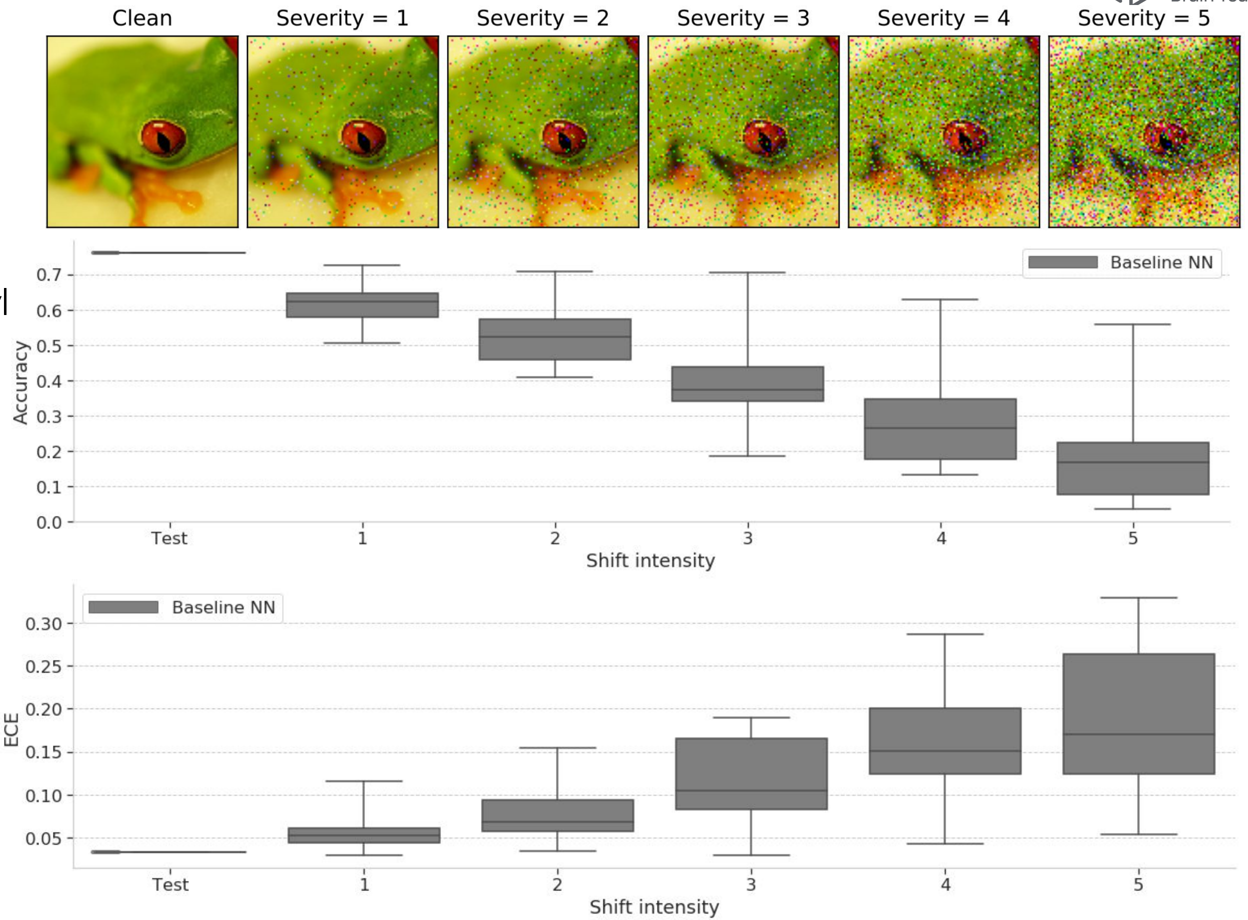
- **Expected Calibration error (↓)**

Calibration Error = |Confidence - Accuracy|

predicted probability of correctness

observed frequency of correctness

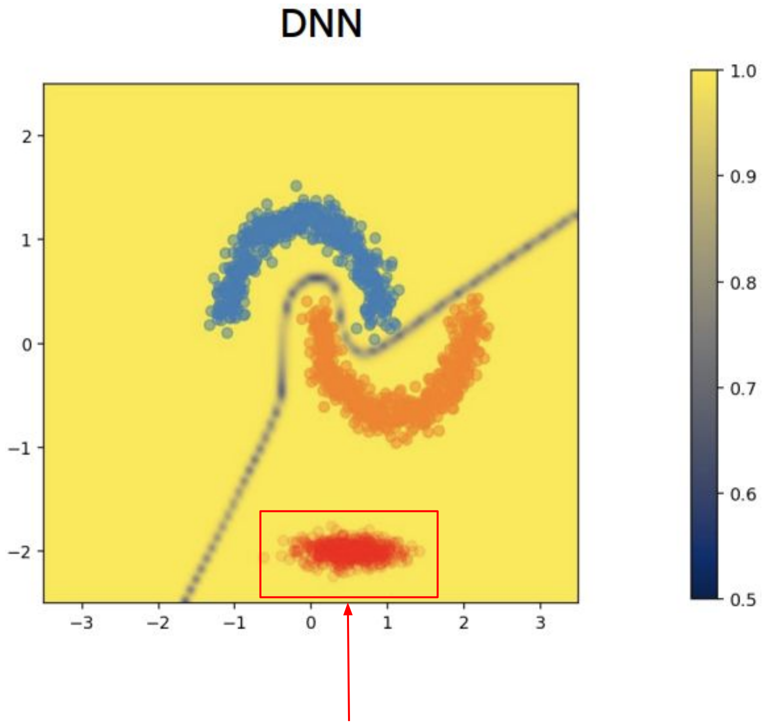
- **Quality of uncertainty degrades with shift ->** "overconfident mistakes"



Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift?, [Ovadia*, Fertig* et al. 2019]

Models assign high confidence predictions far away from training data

Google AI Brain Team



High confidence
(low uncertainty)

Low uncertainty
(high confidence)

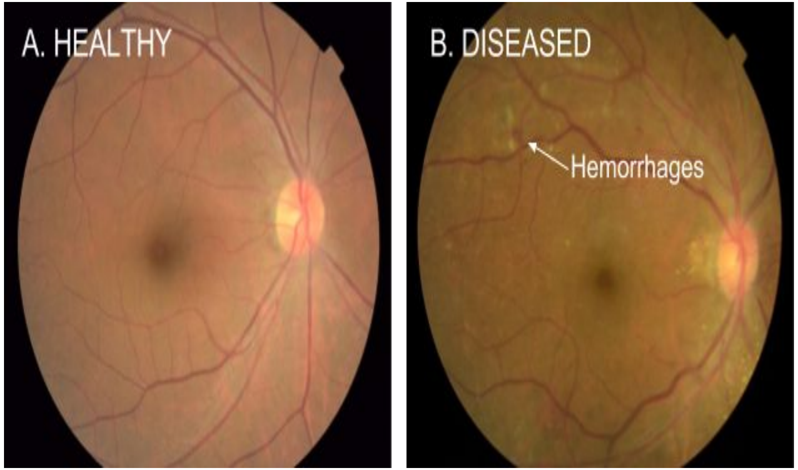
Deep neural networks assign high confidence predictions to inputs far away from $p_{TRAIN}(x,y)$

Applications

Healthcare

Google AI Brain Team

- Use model uncertainty to decide when to trust the model or to defer to a human.
- Selective prediction, Cost-sensitive decision making



Diabetic retinopathy detection from fundus images
[Gulshan et al, 2016]

True label

Action		Healthy	Diseased
	Predict Healthy	0	10
	Predict Diseased	1	0
	Abstain/Defer "I don't know"	0.5	0.5

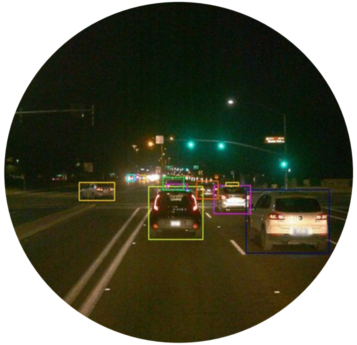
Self-driving cars

Google AI
Brain Team

Dataset shift:

- Time of day / Lighting
- Geographical location (City vs suburban)
- Changing conditions (Weather / Construction)

Daylight



Night

Weather

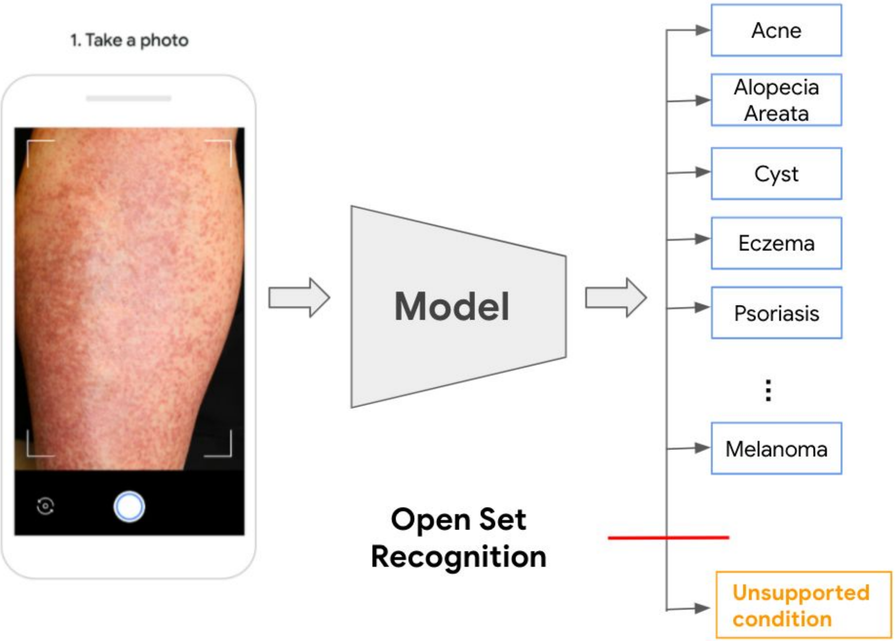
Construction

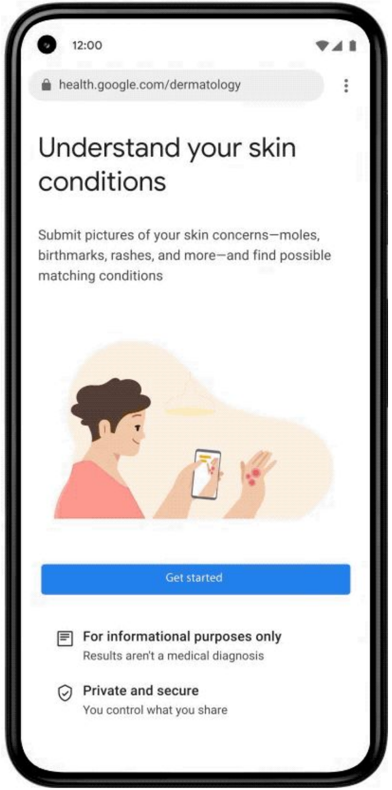
Downtown

Suburban

Image credit: Sun et al, [Waymo Open Dataset]

Open Set Recognition





Test input may not belong to one of the K training classes.

High I.I.D. accuracy is not sufficient, need to be able to detect OOD inputs.

Image source: https://blog.google/technology/health/ai-dermatology-preview-io-2021/

Active Learning

- Use model uncertainty to improve data efficiency and model performance in blindspots.

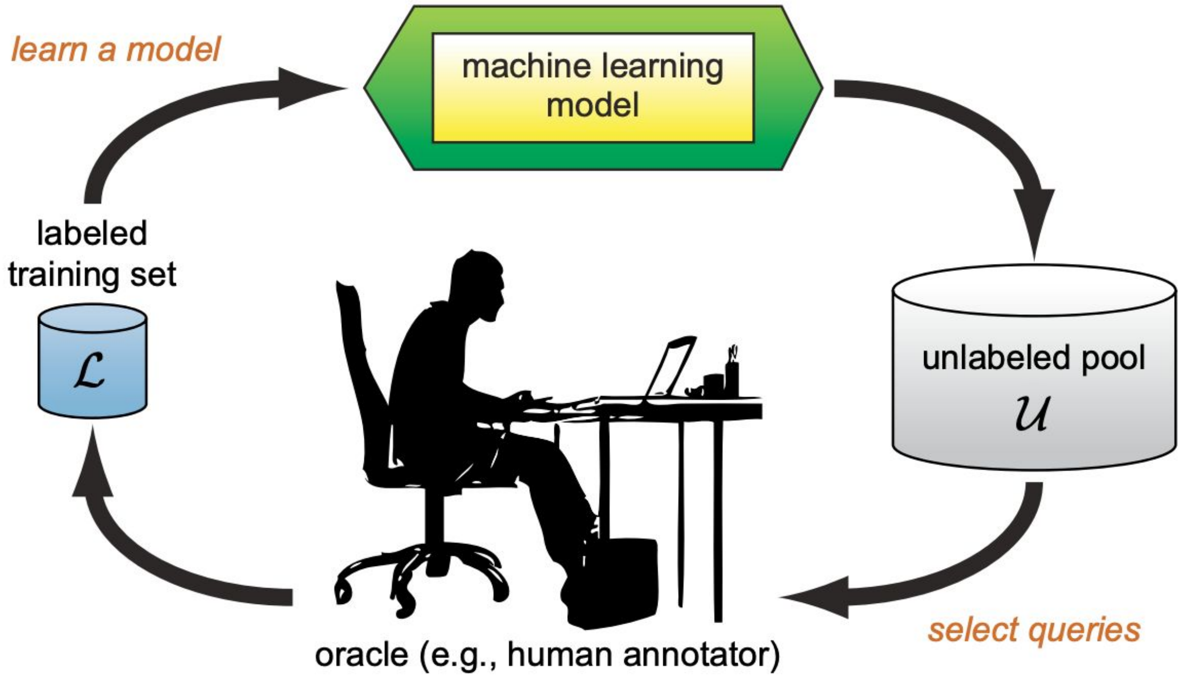


Image source: Active Learning Literature Survey, [Settles 2010]

Bayesian Optimization and Experimental Design

Google AI
Brain Team

- Hyperparameter optimization and experimental design
    - Used across large organizations and the sciences
- Photovoltaics, chemistry experiments, AlphaGo, batteries, materials design

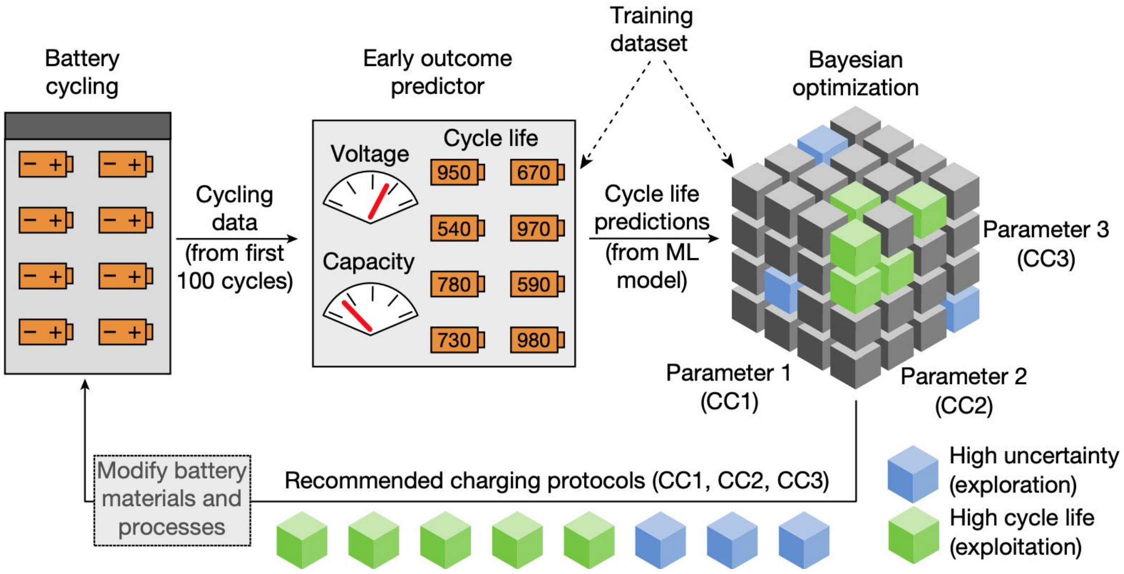


Image source: Attia et al. 2020 Closed-loop optimization of fast-charging protocols for batteries with machine learning

Bandits and Reinforcement Learning

Google AI
Brain Team

- Decision making with asymmetric losses

$$\ell(\mu) \neq \mathbb{E}_{z\sim N(\mu,\sigma^2)}[\ell(z)]$$

- Modeling uncertainty is crucial for **exploration vs exploitation** trade-off

- Non-stationarity

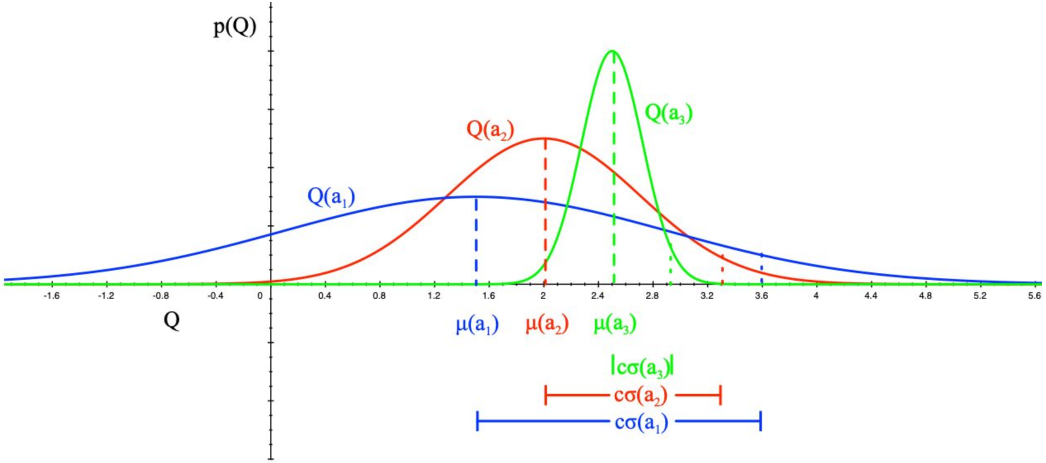
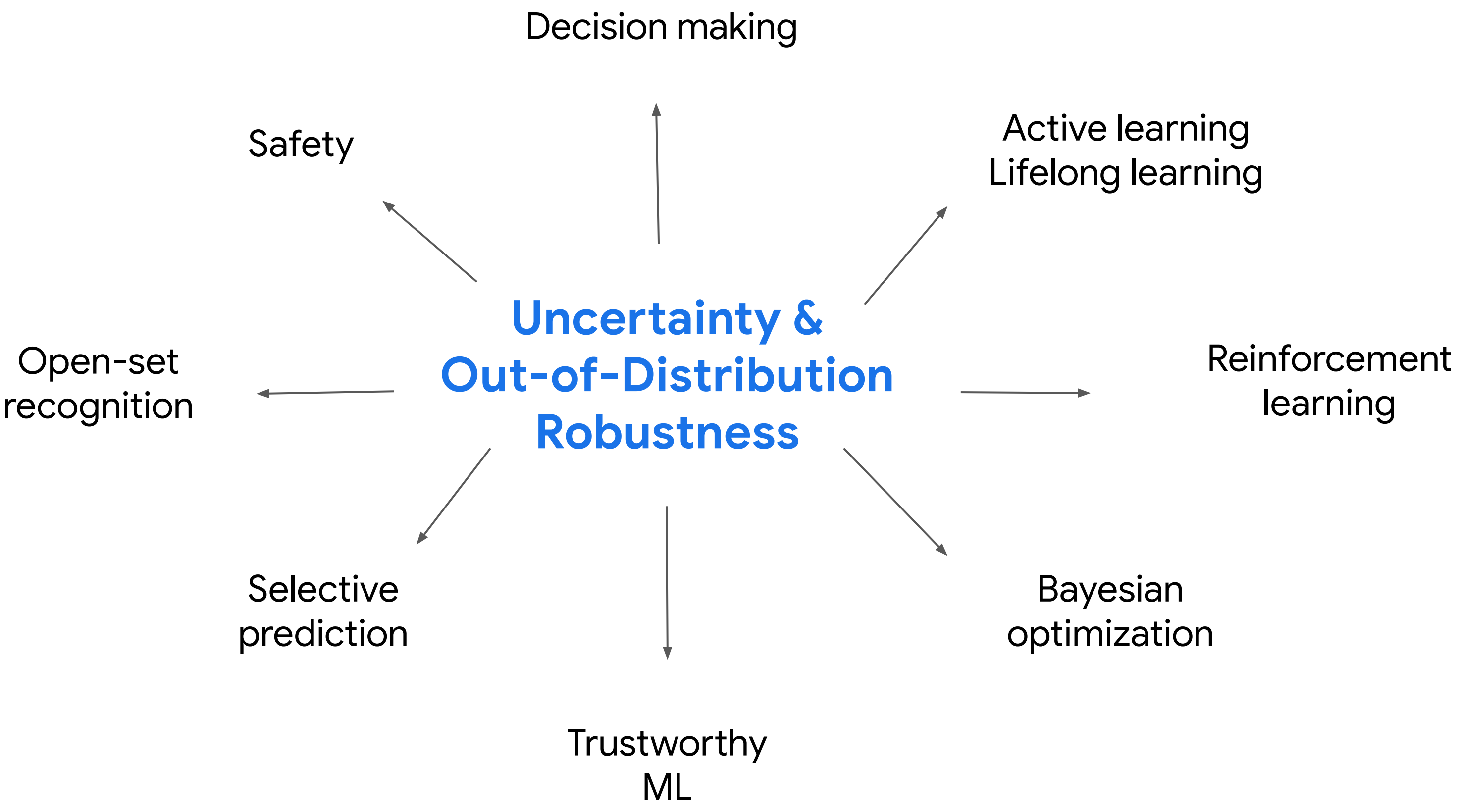


Image source: David Silver's [RL course]

All models are wrong, but ~~some~~ *models that know when they are wrong*, are useful.

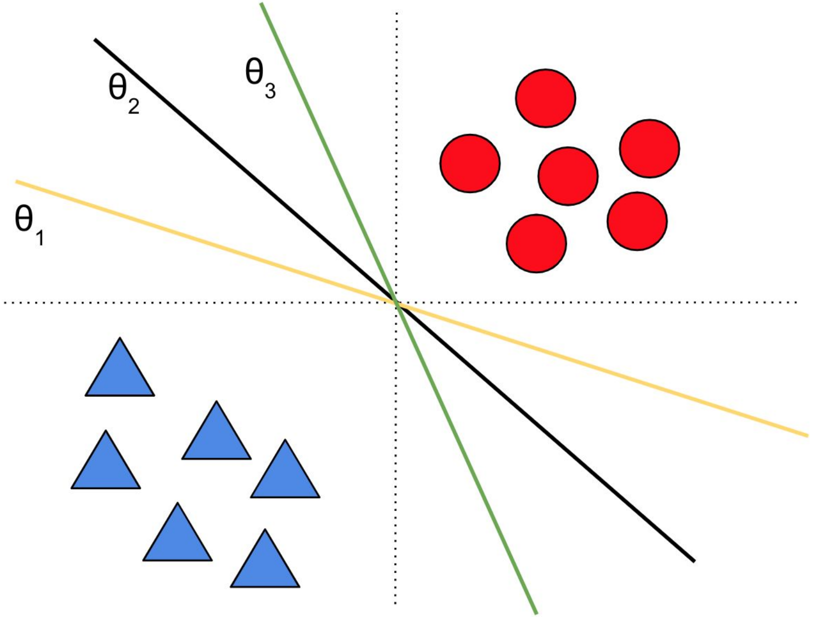


Primer on Uncertainty & Robustness

Sources of uncertainty: *Model uncertainty*

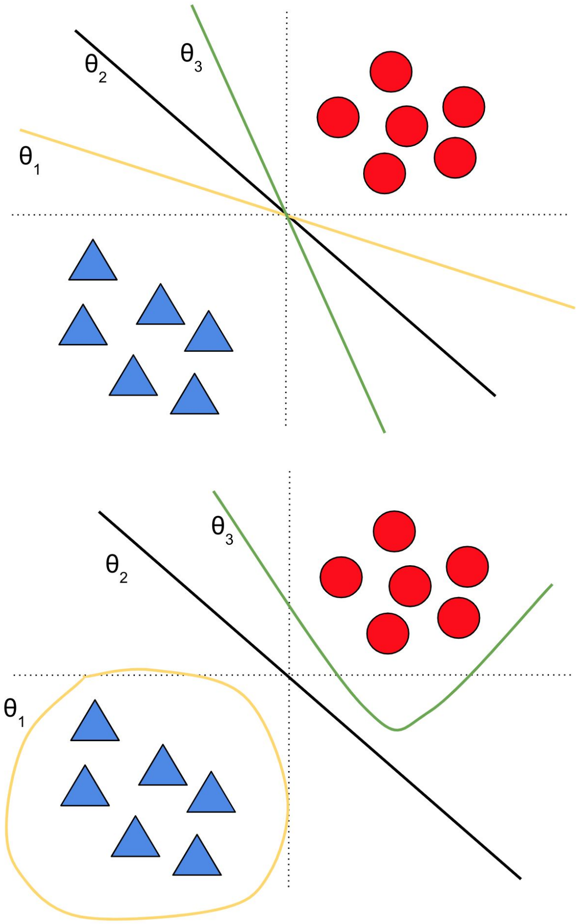
Google AI Brain Team

- Many models can fit the training data well
- Also known as ***epistemic uncertainty***
- Model uncertainty is "**reducible**"
  - Vanishes in the limit of infinite data (subject to model identifiability)



Sources of uncertainty: *Model uncertainty*

- Many models can fit the training data well
- Also known as ***epistemic uncertainty***
- Model uncertainty is "**reducible**"
    - Vanishes in the limit of infinite data (subject to model identifiability)
- Models can be from same hypotheses class (e.g. linear classifiers in top figure) or belong to different hypotheses classes (bottom figure).



Sources of uncertainty: *Data uncertainty*

- Labeling noise (ex: human disagreement)

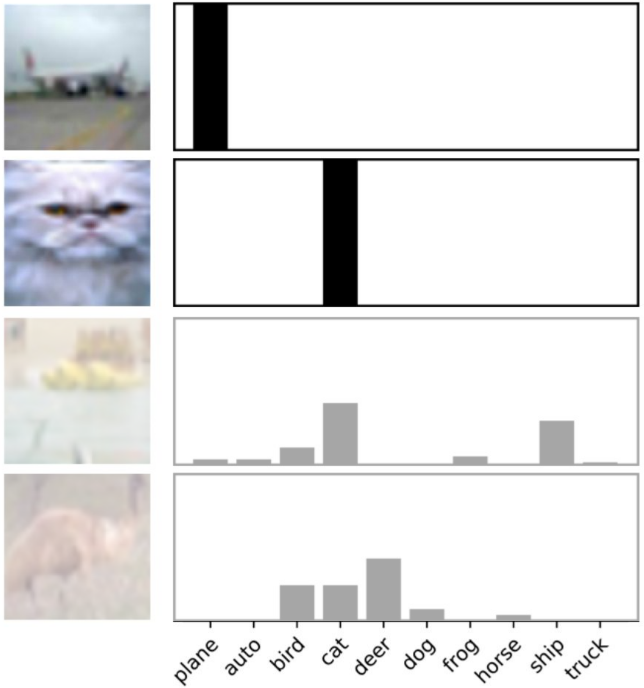


Image source: [Battleday et al. 2019](#) “Improving machine classification using human uncertainty measurements”

Sources of uncertainty: *Data uncertainty*

- Labeling noise (ex: human disagreement)

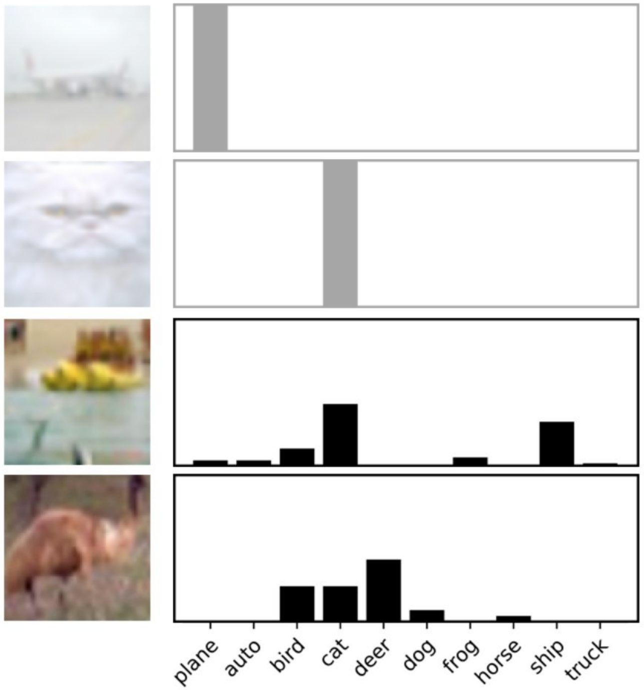


Image source: Battleday et al. 2019 "Improving machine classification using human uncertainty measurements"

Sources of uncertainty: *Data uncertainty*

- Labeling noise (ex: human disagreement)
- Measurement noise (ex: imprecise tools)
- *Missing* data (ex: partially observed features, unobserved confounders)
- Also known as ***aleatoric uncertainty***
- Data uncertainty is "**irreducible***"
  - Persists even in the limit of infinite data
  - *Could be reduced with additional features/views

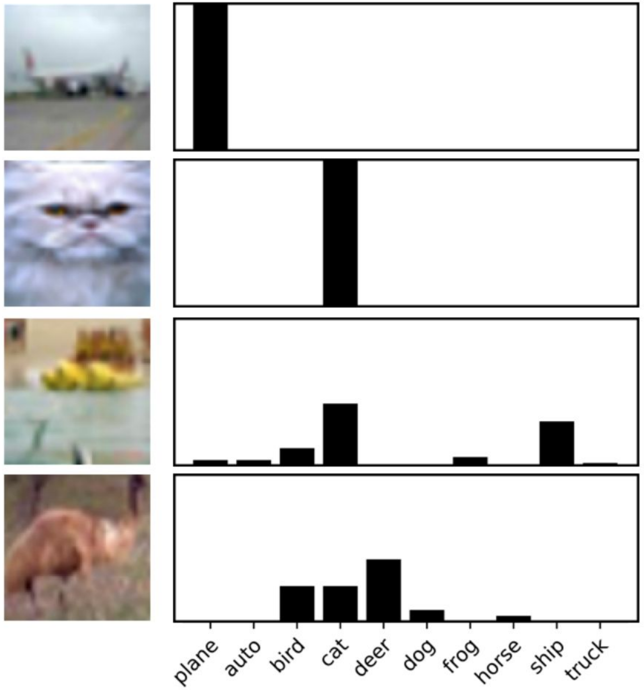


Image source: Battleday et al. 2019 "Improving machine classification using human uncertainty measurements"

How do we measure the quality of uncertainty?

Calibration Error = |Confidence - Accuracy|

predicted probability of correctness

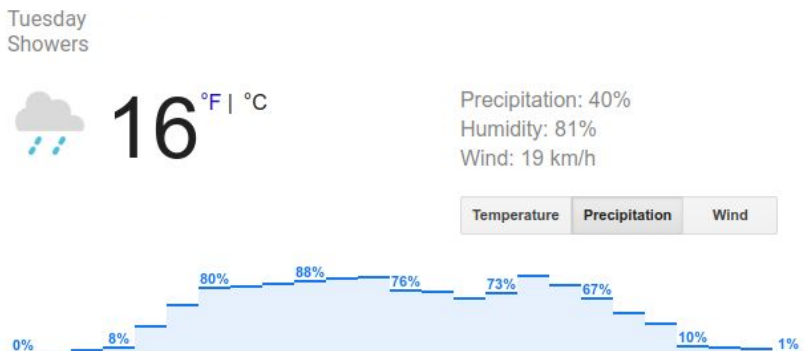
observed frequency of correctness

How do we measure the quality of uncertainty?

Calibration Error = |Confidence - Accuracy|

Of all the days where the model predicted rain with 80% probability, what fraction did we observe rain?

- 80% implies perfect calibration
- Less than 80% implies model is overconfident
- Greater than 80% implies model is under-confident



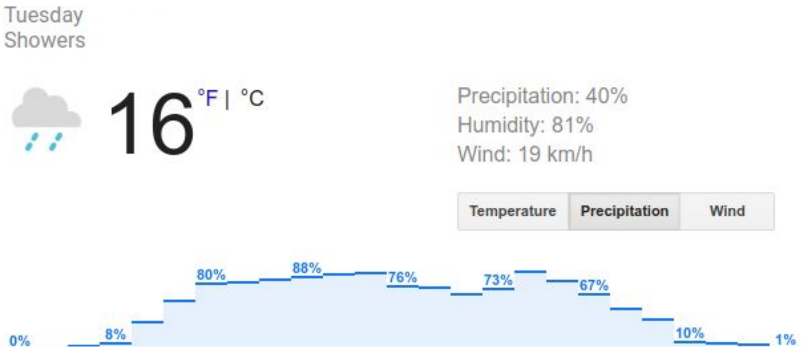
Google AI
Brain Team

How do we measure the quality of uncertainty?

Calibration Error = |Confidence - Accuracy|

Of all the days where the model predicted rain with 80% probability, what fraction did we observe rain?

- 80% implies perfect calibration
- Less than 80% implies model is overconfident
- Greater than 80% implies model is under-confident



Intuition: For regression, calibration corresponds to coverage in a confidence interval.

How do we measure the quality of uncertainty?

Expected Calibration Error [Naeini+ 2015]:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left|\mathrm{acc}(b) - \mathrm{conf}(b)\right|$$

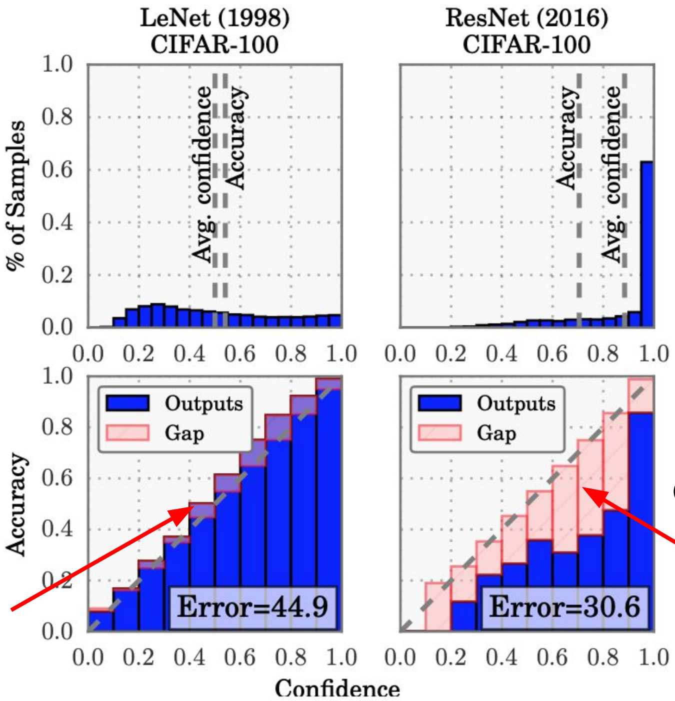
- Bin the probabilities into B bins.
- Compute the within-bin accuracy and within-bin predicted confidence.
- Average the calibration error across bins (weighted by number of points in each bin).

Google AI
Brain Team

How do we measure the quality of uncertainty?

Expected Calibration Error [Naeini+ 2015]:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left|\mathrm{acc}(b) - \mathrm{conf}(b)\right|$$



Confidence < Accuracy

=> Underconfident

Confidence > Accuracy

=> Overconfident

Image source: Guo+ 2017 “On calibration of modern neural networks”

How do we measure the quality of uncertainty?

Expected Calibration Error [Naeini+ 2015]:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left|\mathrm{acc}(b) - \mathrm{conf}(b)\right|$$

Note: Does **not** reflect **accuracy**.

Predicting class frequency p(y=1) = 0.3 for all the inputs achieves perfect calibration.

True label	0	0	0	0	0	0	0	1	1	1	Accurate?	Calibrated?
Model prediction	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	❌	✅

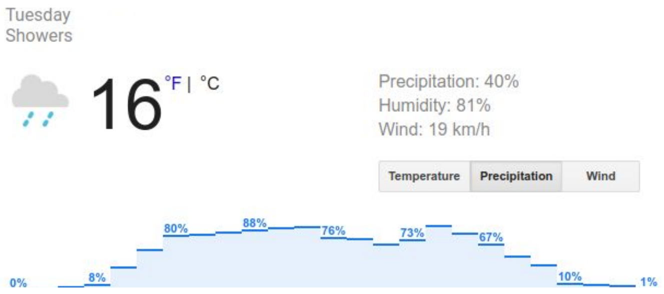
How do we measure the quality of uncertainty?

Proper scoring rules [Gneiting & Raftery 2007]

- Negative Log-Likelihood (NLL)
  - Also known as *cross-entropy*
  - Can overemphasize tail probabilities
- Brier Score
  - Quadratic penalty (bounded range [0,1] unlike log).

$$\mathrm{BS} = \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}}\left[p(y|\mathbf{x}_n,\theta)-\delta(y-y_n)\right]^2$$

  - Can be numerically unstable to optimize.

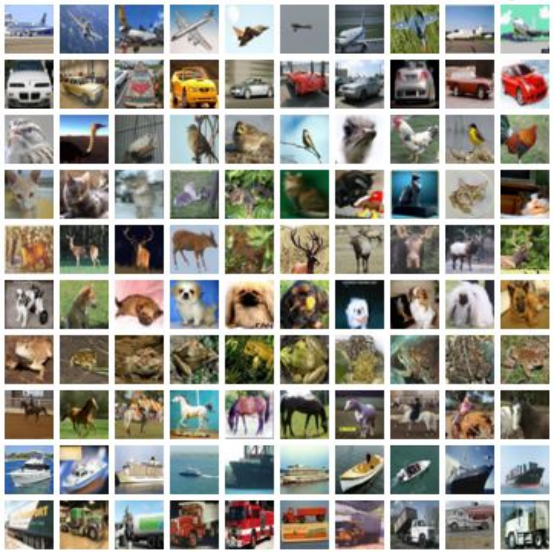


How do we measure the quality of uncertainty?

Google AI
Brain Team

Evaluate model on **out-of-distribution (OOD) inputs** which do not belong to any of the existing classes

- Max confidence
- Entropy of p(y|x)



CIFAR-10 (IID test inputs)

SVHN (OOD test inputs)

CIFAR-10 classifier

Confidence on IID inputs > Confidence on OOD inputs ?

Overview of Methods

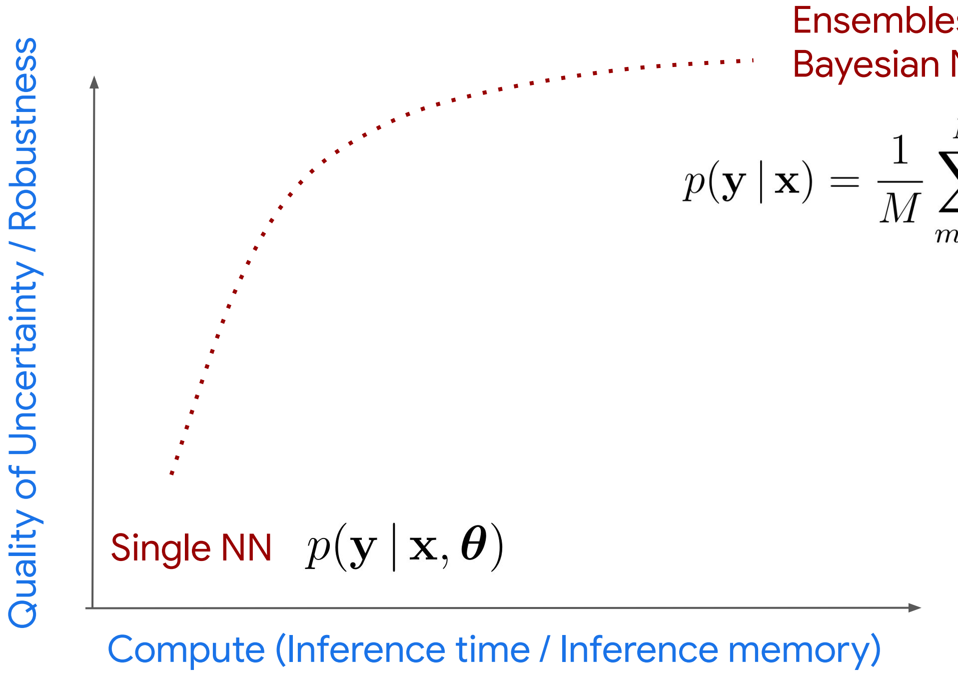
Probabilistic Deep Learning

Google AI Brain Team

- Parametrize "base model". $$p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$$
- Specify prior over functions. $$p(\boldsymbol{\theta})$$
- Capture model uncertainty by approximating the posterior. $$p(\boldsymbol{\theta} \mid \mathcal{D})$$
- Average predictions over multiple functions (ensemble or Bayesian NN) $$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_m)$$

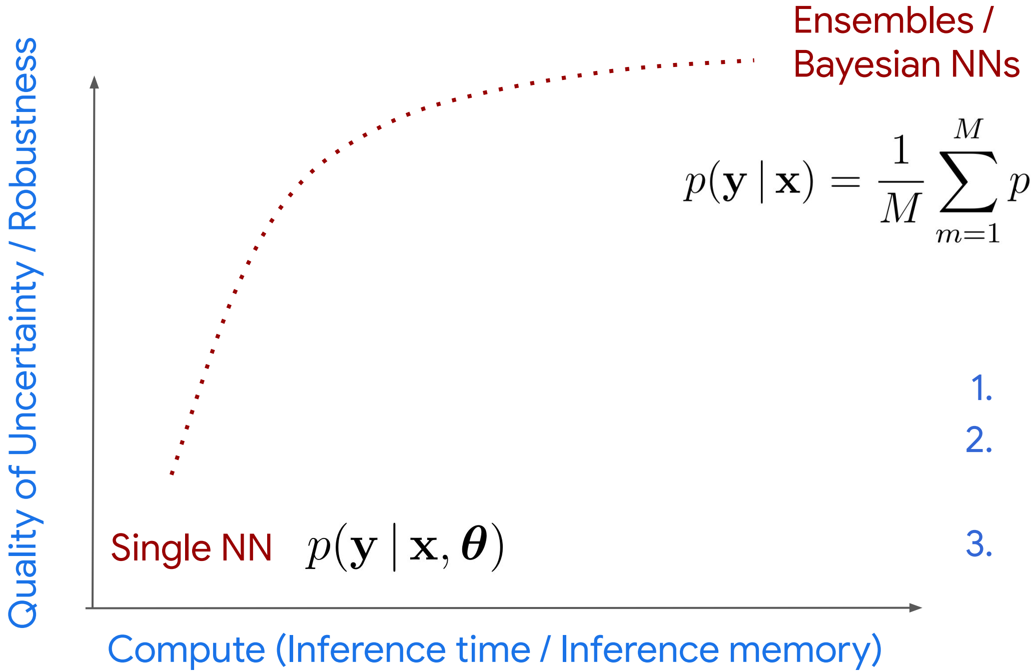
Cartoon: Uncertainty/Robustness vs Compute frontier

Google AI
Brain Team



- Probabilistic framework gives an unifying view. Improving p(y|x) improves performance on **all** downstream tasks (accuracy/calibration under shift, selective prediction, open set recognition, etc) as opposed to custom techniques for tasks.
- Practitioners can pick “operating point” depending on constraints of application.

Orthogonal ways of improving performance



$$p(\mathbf{y} \,|\, \mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{y} \,|\, \mathbf{x}, \boldsymbol{\theta}_m)$$

$$p(\mathbf{y} \,|\, \mathbf{x}, \boldsymbol{\theta})$$

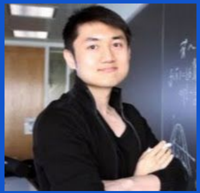
“Composable” toolkit

1. Improve the “base” model p(y|x,θ)
2. Efficiently average predictions over diverse set of functions $\theta_1, \theta_2 \dots \theta_M$
3. Better inductive biases for representations (e.g. pre-training or data augmentation)

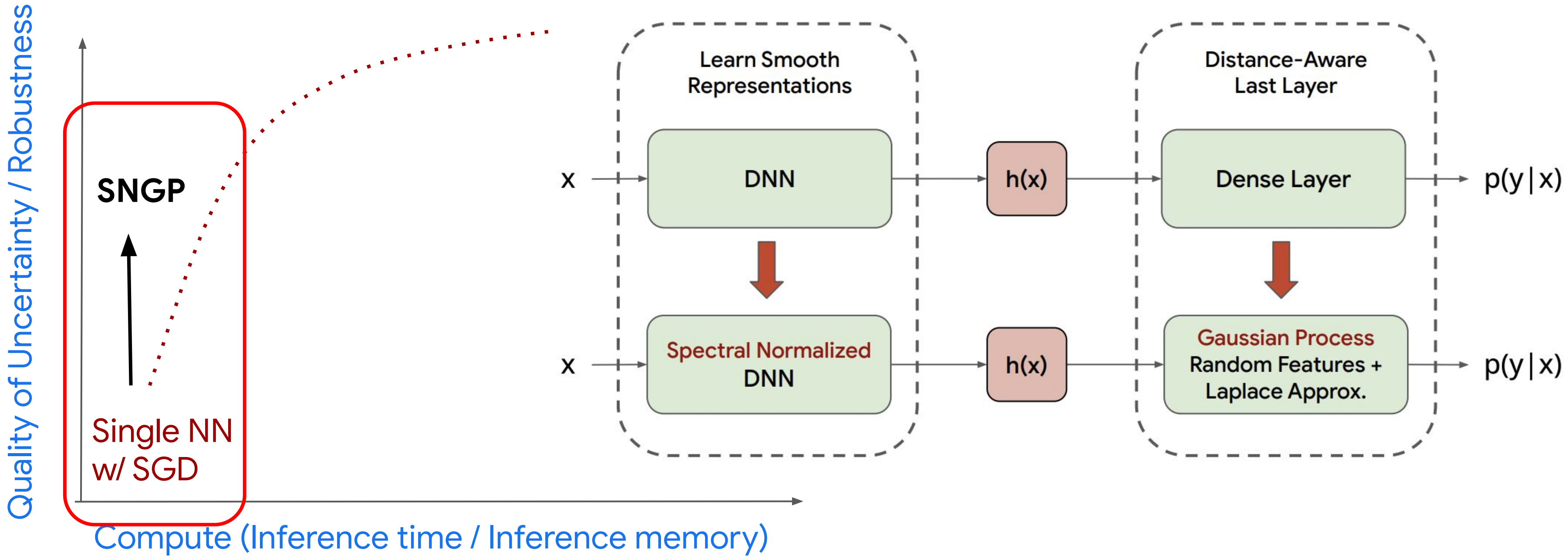
Composing can further improve performance!

Improving Single Model Uncertainty via Distance Awareness

Jeremiah Liu*, Shreyas Padhy*, Jie Ren*, et al.

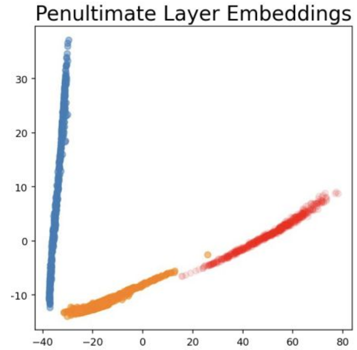


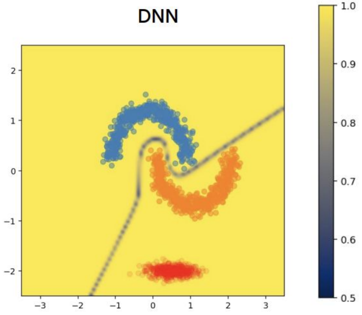
SNGP improves single model uncertainty with two simple changes:
Spectral-normalization (SN) + Last-layer Gaussian Process (GP)

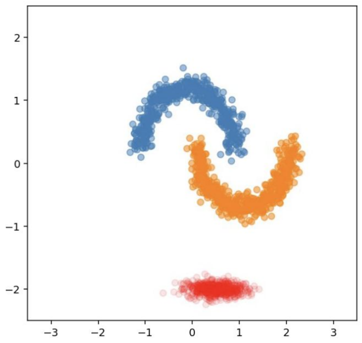


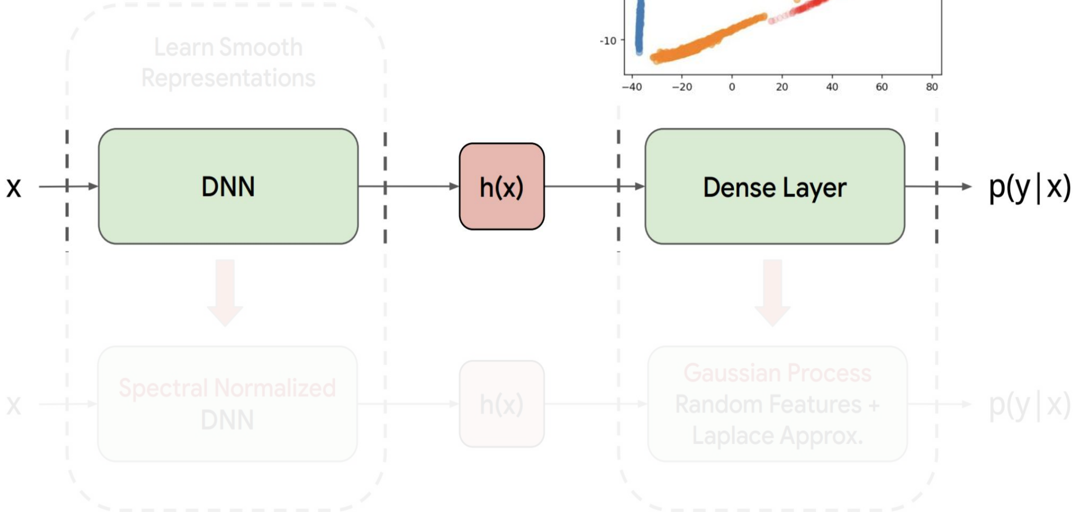
Google AI
Brain Team

Why do vanilla DNNs assign high confidence predictions far away from training data?







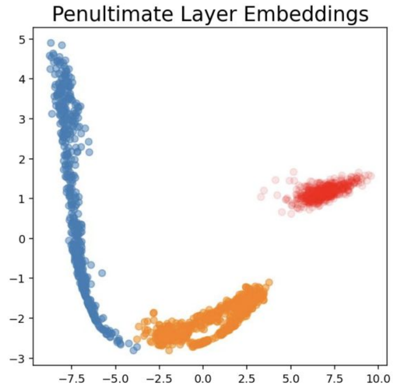


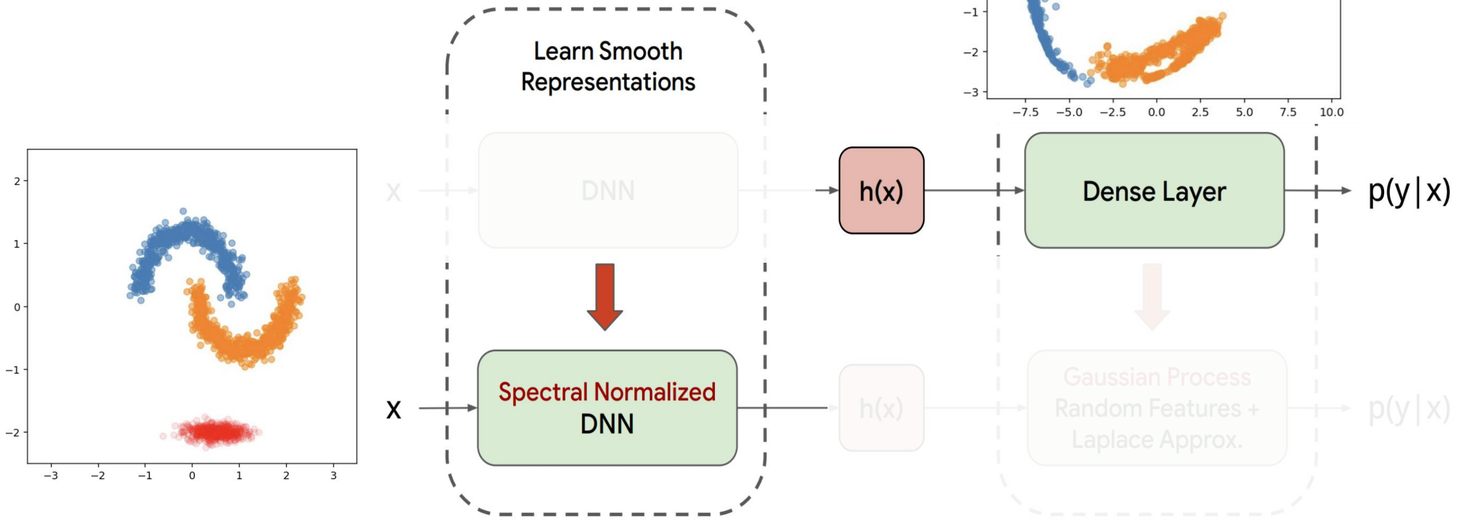
1. Vanilla NNs can map inputs far away in input space to close points in latent space (cf. “shortcut learning”).

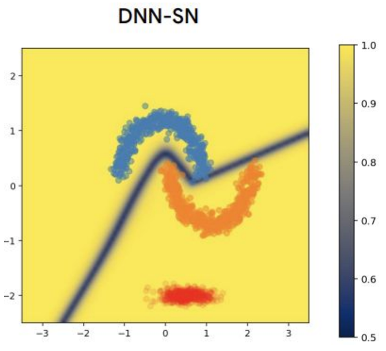
2. Confidence is a function of distance from boundary (and not the training data).

Google AI
Brain Team

Idea 1: Enforce bi-Lipschitz smoothness via spectral normalization





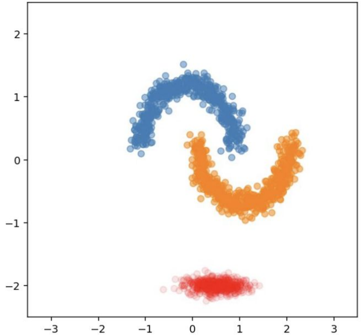


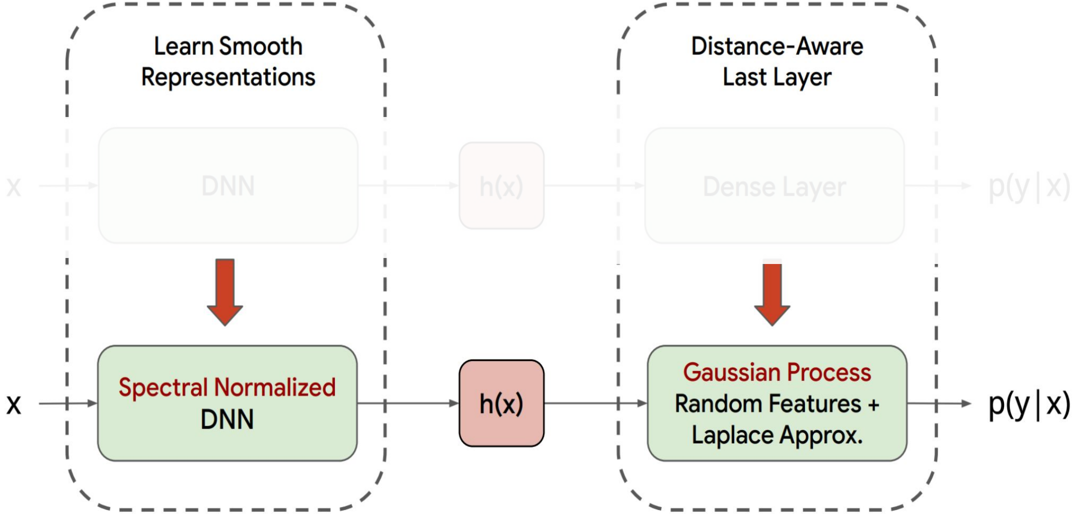
Bi-Lipschitz smoothness discourages inputs far away in input space getting mapped close in latent space.

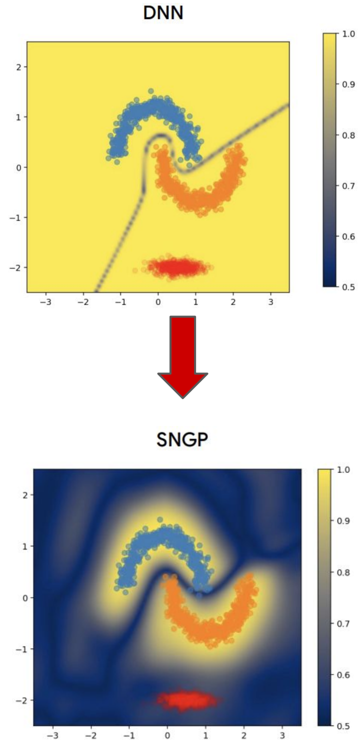
Google AI
Brain Team

Idea 1: Enforce bi-Lipschitz smoothness

Idea 2: Replace last dense layer with “Gaussian process” layer







Bi-Lipschitz smoothness discourages inputs far away in input space getting mapped close in latent space

For GP layer, confidence is a function of distance from the training data.

Spectral-normalized Neural Gaussian process (SNGP)

Google AI Brain Team

- SNGP improves
  - Accuracy under shift
  - Calibration under shift
  - OOD detection
- Simple to implement
- Low computational/memory overhead
- A building block for better ensembles

Method	Accuracy (↑)	ECE (↓)	OOD AUROC (↑)	OOD AUPR (↑)	Latency (ms / example)
Deterministic	96.5	0.0236	0.8970	0.7573	**10.42**
MCD-GP	95.9	0.0146	0.9055	0.8030	88.38
DUQ	96.0	0.0585	0.9173	0.8058	15.60
MC Dropout	96.5	0.0210	0.9382	0.7997	85.62
Deep Ensemble	**97.5**	0.0128	0.9635	0.8616	84.46
SNGP	96.6	**0.0115**	**0.9688**	**0.8802**	17.36

BERT on an intent detection benchmark

Method	Acc (↑)	ECE (↓)	Corrupted Acc/ECE	OOD AUPR (↑) SVHN	OOD AUPR (↑) CIFAR100	Latency (ms)
Deterministic	96.0	0.023	72.9 / 0.153	0.7810	0.8352	**3.91**
MCD-GP	95.5	0.024	70.0 / 0.100	0.9599	0.8631	29.53
DUQ	94.7	0.034	71.6 / 0.183	0.9733	0.8537	8.68
MC Dropout	96.0	0.024	70.0 / 0.116	0.9714	0.8320	27.10
Deep Ensembles	**96.6**	**0.010**	**77.9 / 0.087**	0.9640	0.8875	38.10
SNGP (Ours)	95.9	0.018	74.6 / 0.090	**0.9901**	**0.9050**	6.25

Results on CIFAR-10 using Wide ResNet

See also [van Amersfoort+ 2020].

[Liu+ 2020]

SNGP is very easy to implement

Google AI Brain Team

Algorithm 1 SNGP Training

1: **Input:**
Minibatches $\{D_i\}_{i=1}^N$ for $D_i = \{y_m, \mathbf{x}_m\}_{m=1}^M$.

2: **Initialize:**

$$\hat{\Sigma} = \tau * \mathbf{I}, \mathbf{W}_L \overset{iid}{\sim} N(0,1), \mathbf{b}_L \overset{iid}{\sim} U(0, 2\pi)$$

3: **for** train_step $= 1$ **to** max_step **do**
4: SGD update $\left\{\beta, \{\mathbf{W}_l\}_{l=1}^{L-1}, \{\mathbf{b}_l\}_{l=1}^{L-1}\right\}$ (12)
5: **if** final_epoch **then**
6: Update precision matrix $\hat{\Sigma}^{-1}$ (11).
7: **end if**
8: **end for**
9: Compute posterior covariance $\hat{\Sigma} = \text{inv}(\hat{\Sigma}^{-1})$.

Algorithm 2 SNGP Prediction

1: **Input:** Testing example $\mathbf{x}$.
2: Compute Features:

$$\Phi_{D_L \times 1} = \sqrt{2\sigma^2/D_L} * \cos(\mathbf{W}_L h(\mathbf{x}) + \mathbf{b}_L)$$

3: Compute Posterior Mean:

$$\text{logit}(\mathbf{x}) = \Phi^\top \beta$$

4: Compute Posterior Variance:

$$\text{var}(\mathbf{x}) = \Phi^\top \hat{\Sigma} \Phi.$$

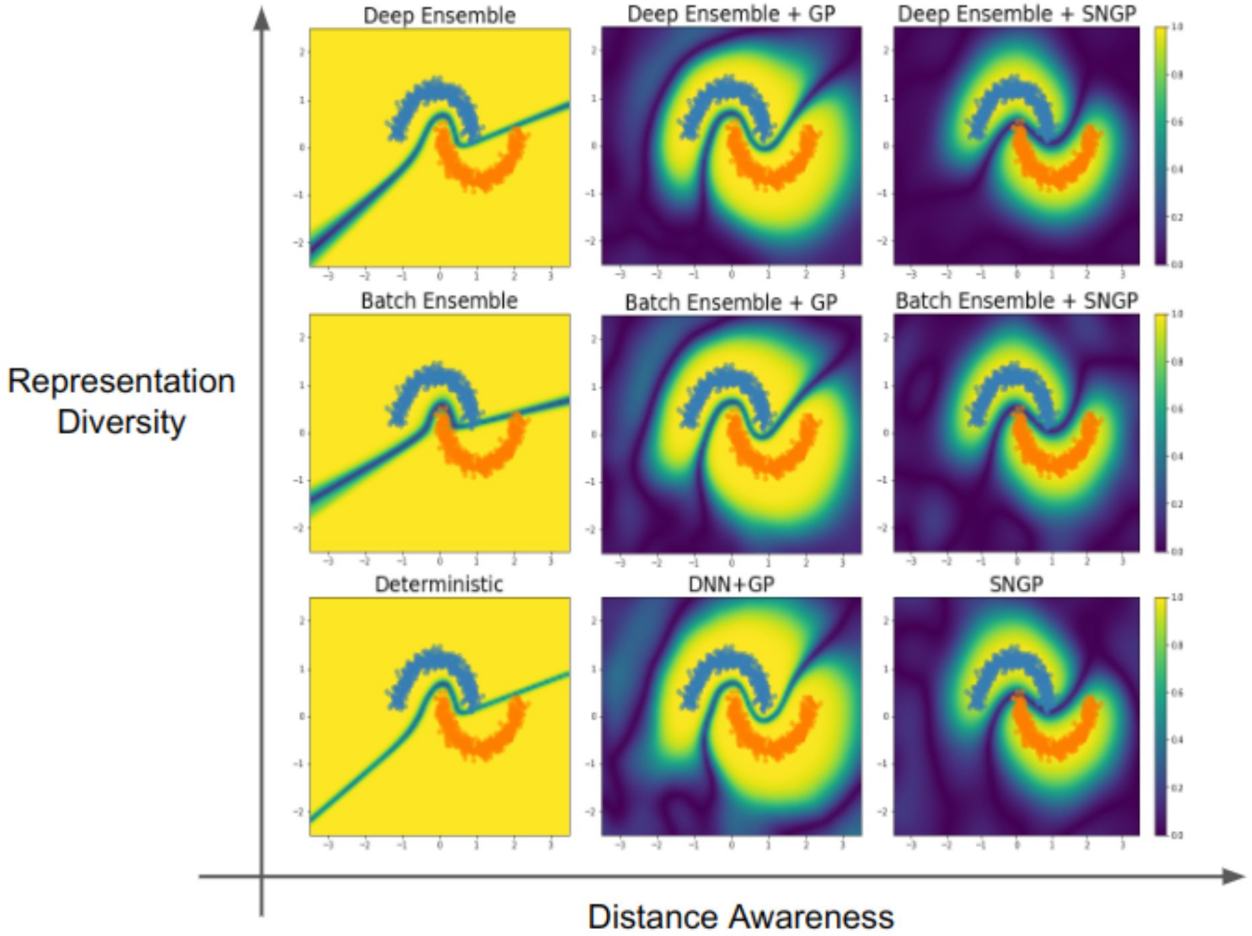
5: Compute Predictive posterior distribution:

$$p(y|\mathbf{x}) = \int_{g \sim N(\text{logit}(\mathbf{x}), \text{var}(\mathbf{x}))} \text{sigmoid}(g) dg$$

[Liu*, Padhy*, Ren* et al. 2022]

SNGP provides complementary benefits to ensembling

Google AI Brain Team



[Liu*, Padhy*, Ren* et al. 2022]

SNGP provides complementary benefits to data augmentation

Method	CIFAR-100 Acc / cAcc (↑)	ECE / cECE (↓)	AUROC SVHN / CIFAR-10 (↑)
DNN	80.4 ± 0.290 / 55.0 ± 0.180	0.107 ± 0.004 / 0.258 ± 0.004	0.799 ± 0.020 / 0.795 ± 0.001
DNN-SN	**80.5 ± 0.300** / 55.0 ± 0.210	0.111 ± 0.002 / 0.268 ± 0.005	0.798 ± 0.022 / 0.793 ± 0.003
DNN-GP	80.3 ± 0.400 / **55.3 ± 0.300**	0.034 ± 0.005 / **0.059 ± 0.002**	0.835 ± 0.021 / 0.797 ± 0.001
SNGP	80.3 ± 0.230 / **55.3 ± 0.190**	**0.030 ± 0.004** / 0.060 ± 0.004	**0.846± 0.019 / 0.798 ± 0.001**
DNN + AugMix	81.6 ± 0.003 / **66.4 ± 0.280**	0.082 ± 0.003 / 0.131 ± 0.005	0.814 ± 0.025 / **0.798 ± 0.003**
DNN-SN + AugMix	81.9 ± 0.280 / **66.4 ± 0.240**	0.080 ± 0.002 / 0.133 ± 0.004	0.824 ± 0.021 / 0.796 ± 0.002
DNN-GP + AugMix	81.6 ± 0.350 / 66.2 ± 0.220	**0.042 ± 0.004** / 0.066 ± 0.002	0.855 ± 0.019 / 0.797 ± 0.002
SNGP + AugMix	81.6 ± 0.240 / **66.4 ± 0.190**	**0.042 ± 0.004 / 0.064 ± 0.002**	**0.870 ± 0.024 / 0.798 ± 0.001**

[Liu*, Padhy*, Ren* et al. 2022]

SNGP scales well to ImageNet

Google AI Brain Team

Method	Accuracy (↑) Clean	Accuracy (↑) Corrupted	ECE (↓) Clean	ECE (↓) Corrupted	NLL (↓) Clean	NLL (↓) Corrupted
Single Model						
DNN	76.2 ± 0.01	40.5 ± 0.01	0.032 ± 0.002	0.103 ± 0.011	0.939 ± 0.01	3.21 ± 0.02
DNN-SN	$\mathbf{76.4 \pm 0.01}$	40.6 ± 0.01	0.079 ± 0.001	0.074 ± 0.001	0.96 ± 0.01	3.14 ± 0.02
DNN-GP	76.0 ± 0.01	$\mathbf{41.3 \pm 0.01}$	0.017 ± 0.001	0.049 ± 0.001	0.93 ± 0.01	3.06 ± 0.02
SNGP (Ours)	76.1 ± 0.01	41.1 ± 0.01	$\mathbf{0.013 \pm 0.001}$	$\mathbf{0.045 \pm 0.012}$	$\mathbf{0.93 \pm 0.01}$	$\mathbf{3.03 \pm 0.01}$
Ensemble Model						
MC Dropout	76.6 ± 0.01	42.4 ± 0.02	0.026 ± 0.002	$\mathbf{0.046 \pm 0.009}$	0.919 ± 0.01	2.96 ± 0.01
Deep Ensemble	77.9 ± 0.01	$\mathbf{44.9 \pm 0.01}$	$\mathbf{0.017 \pm 0.001}$	0.047 ± 0.004	0.857 ± 0.01	2.82 ± 0.01
SNGP Ensemble (Ours)	$\mathbf{78.1 \pm 0.01}$	$\mathbf{44.9 \pm 0.01}$	0.039 ± 0.001	0.050 ± 0.002	$\mathbf{0.851 \pm 0.01}$	$\mathbf{2.77 \pm 0.01}$

[Liu*, Padhy*, Ren* et al. 2022]

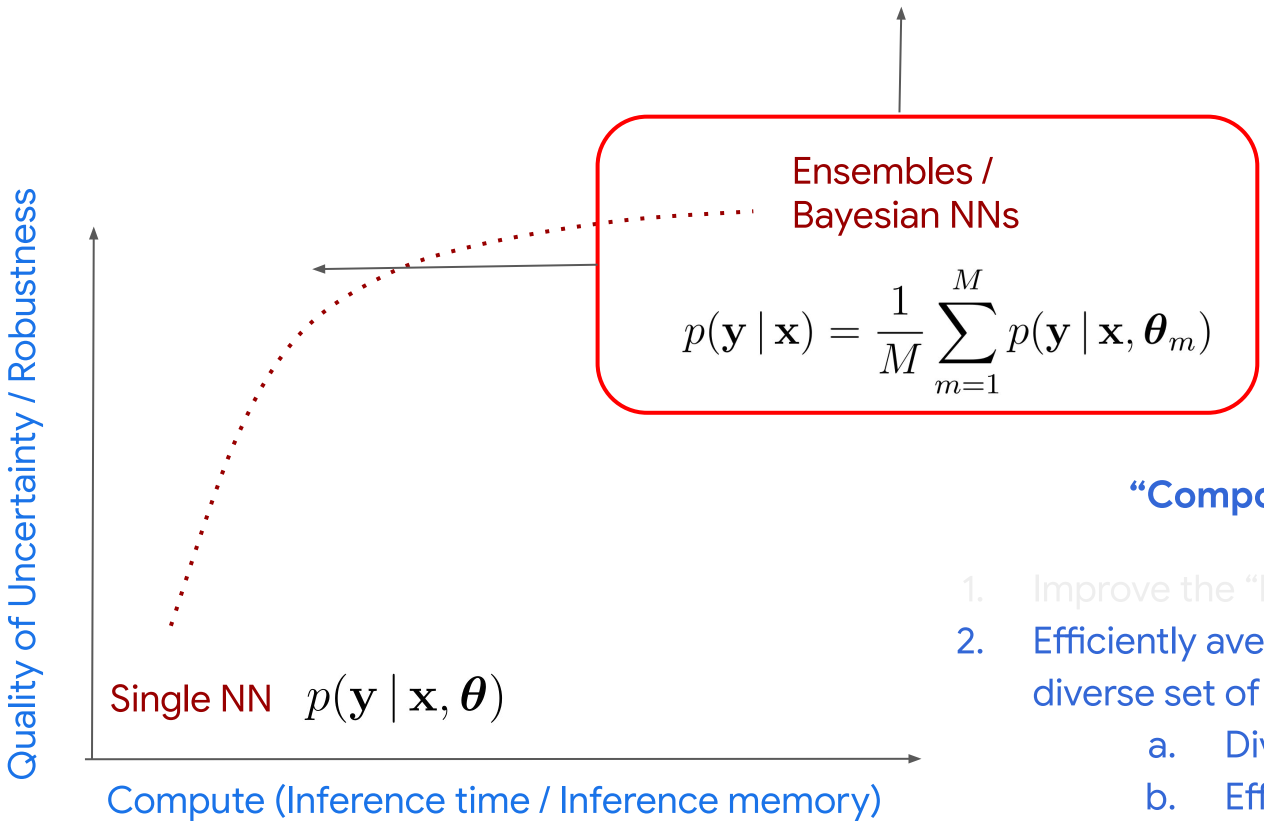
SNGP works well on other modalities, e.g. genomics

Google AI
Brain Team

Method	Accuracy (↑)	ECE (↓)	NLL (↓)	OOD AUROC (↑)	OOD AUPR (↑)
Single Model					
DNN	84.40 ± 0.390	0.049 ± 0.007	0.487 ± 0.007	0.640 ± 0.005	0.609 ± 0.005
DNN-SN	85.56 ± 0.150	0.025 ± 0.003	0.422 ± 0.004	0.658 ± 0.006	0.625 ± 0.005
DNN-GP	85.23 ± 0.200	0.041 ± 0.006	0.457 ± 0.005	0.654 ± 0.006	0.629 ± 0.003
SNGP (Ours)	$\mathbf{85.71} \pm \mathbf{0.100}$	$\mathbf{0.019} \pm \mathbf{0.004}$	$\mathbf{0.417} \pm \mathbf{0.004}$	$\mathbf{0.672} \pm \mathbf{0.011}$	$\mathbf{0.637} \pm \mathbf{0.009}$
Ensemble Model					
MC Dropout	84.16 ± 0.370	0.033 ± 0.003	0.480 ± 0.003	0.641 ± 0.004	0.609 ± 0.004
Deep Ensemble	87.21 ± 0.630	$\mathbf{0.014} \pm \mathbf{0.006}$	0.373 ± 0.012	0.671 ± 0.005	0.640 ± 0.005
SNGP Ensemble (Ours)	$\mathbf{88.19} \pm \mathbf{0.560}$	0.049 ± 0.004	$\mathbf{0.357} \pm \mathbf{0.012}$	$\mathbf{0.687} \pm \mathbf{0.008}$	$\mathbf{0.656} \pm \mathbf{0.007}$

[Liu*, Padhy*, Ren* et al. 2022]

Improving the quality of model uncertainty



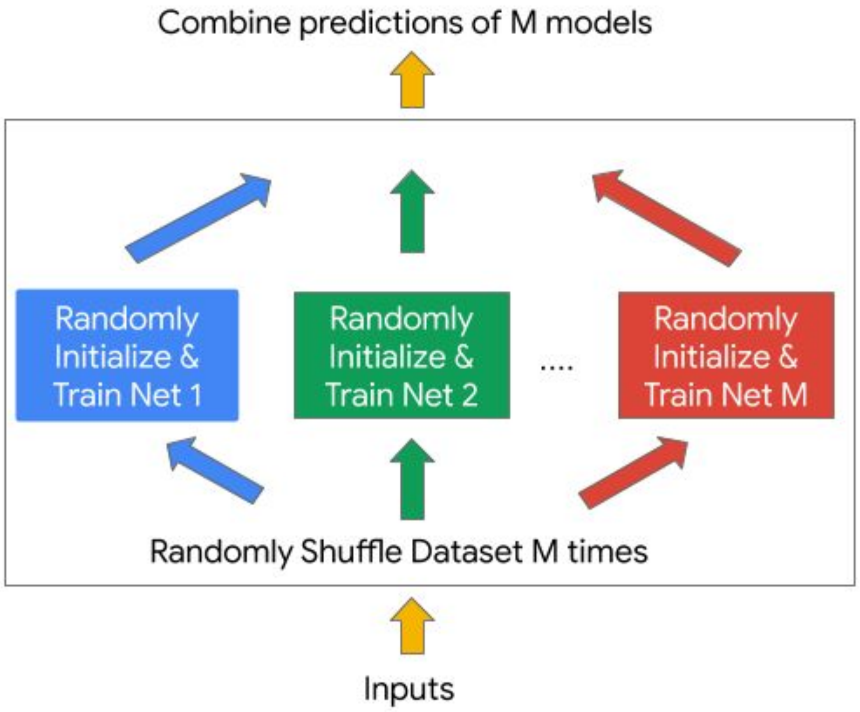
"Composable" toolkit

1. Improve the "base" model p(y|x,θ)
2. Efficiently average predictions over diverse set of functions θ_1, θ_2 ... θ_M
   a. Diverse ensembles ↑
   b. Efficient ensembles ⟵

Surprisingly Simple Baseline: Deep Ensembles

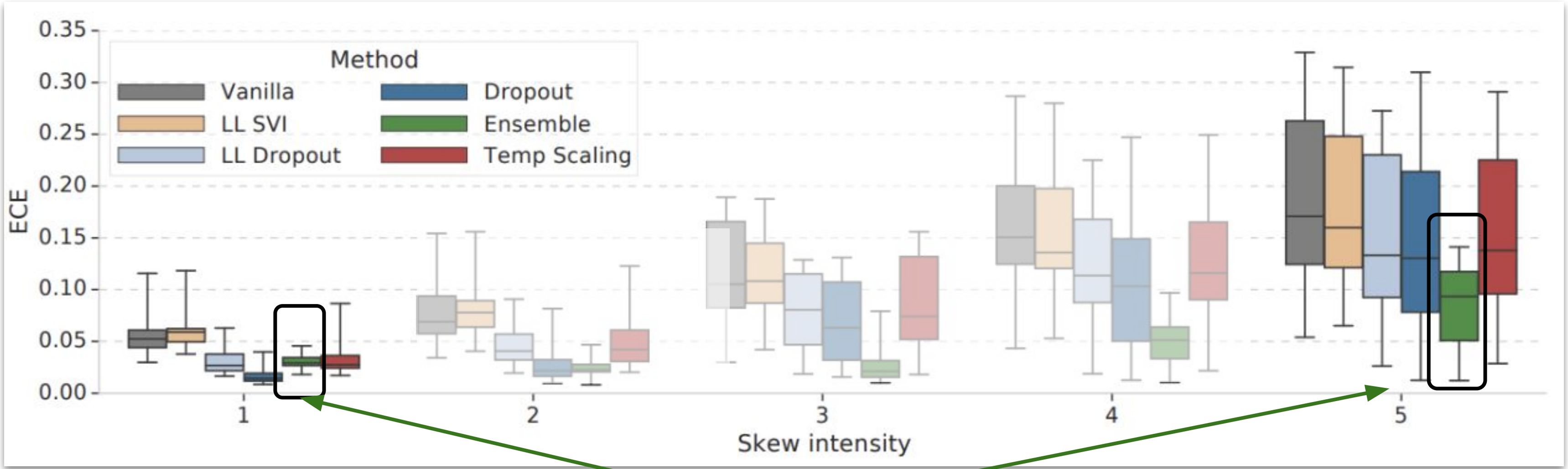
Google AI
Brain Team

Just re-run standard training but with different random seeds (initializations + SGD shuffling) & combine models.



[Lakshminarayanan+ 2016]

Deep Ensembles improve accuracy and calibration under dataset shift

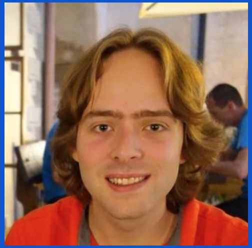


Deep Ensembles are consistently among the best performing methods, especially under dataset shift

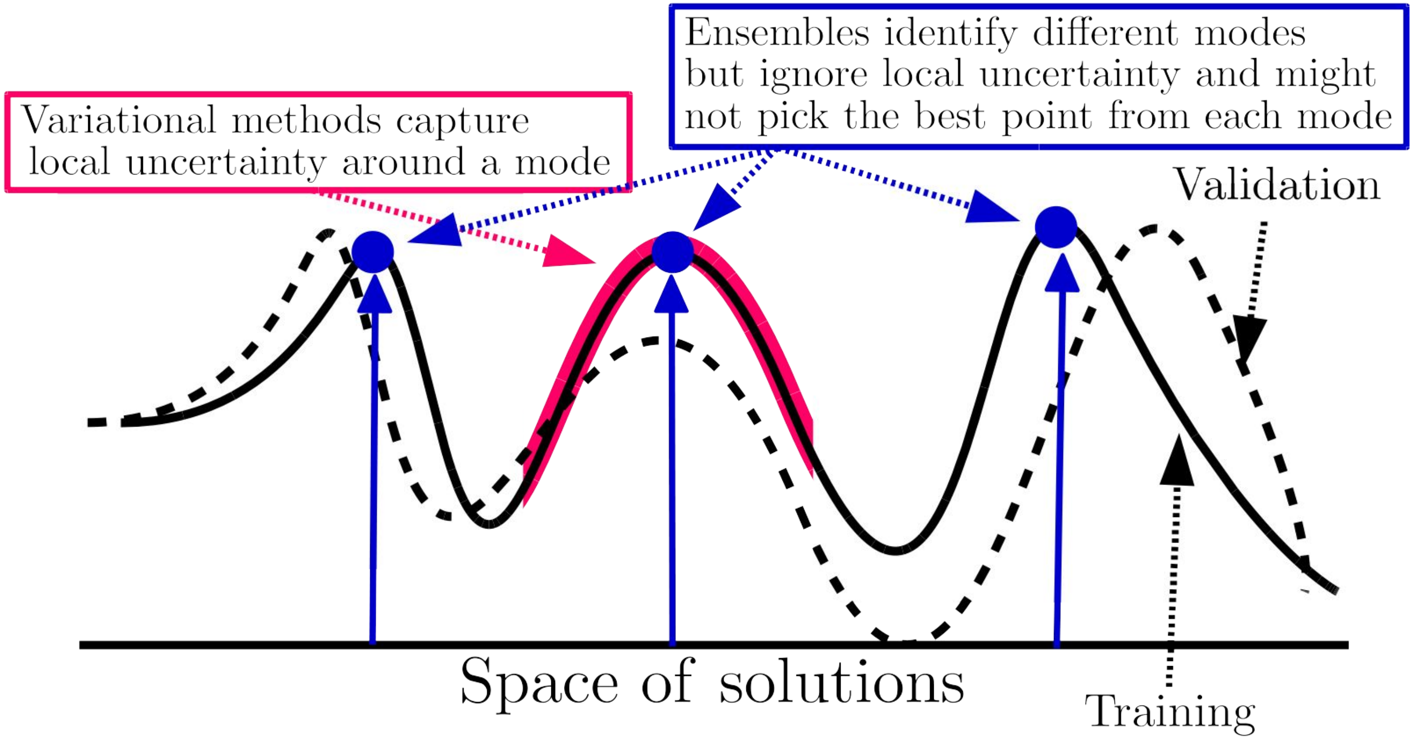
Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift [Ovadia+ 2019]

Deep Ensembles: A Loss Landscape Perspective

Stanislav Fort*, Clara Huiyi Hu*, Balaji Lakshminarayanan




Motivation: Understand why deep ensembles work well

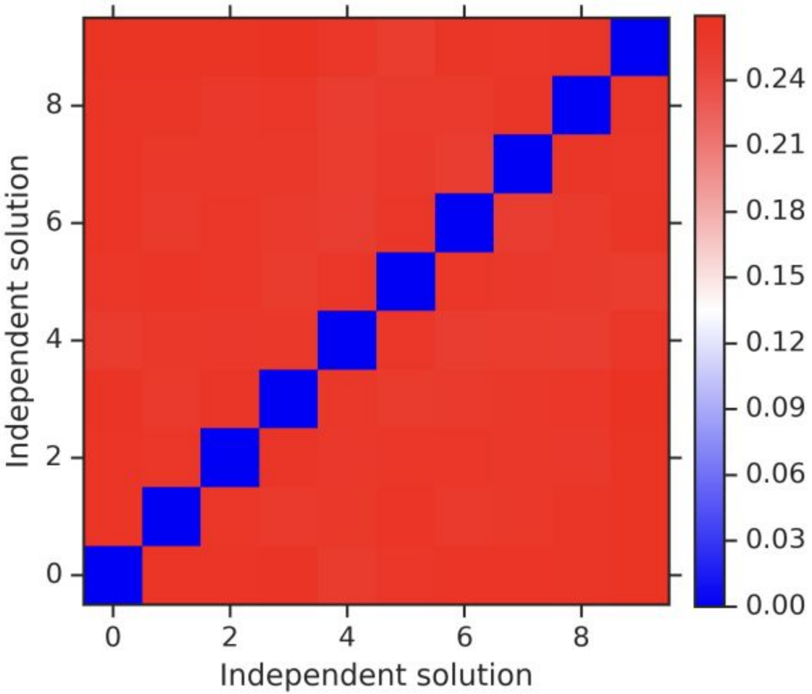


Function space distance = prediction dissimilarity

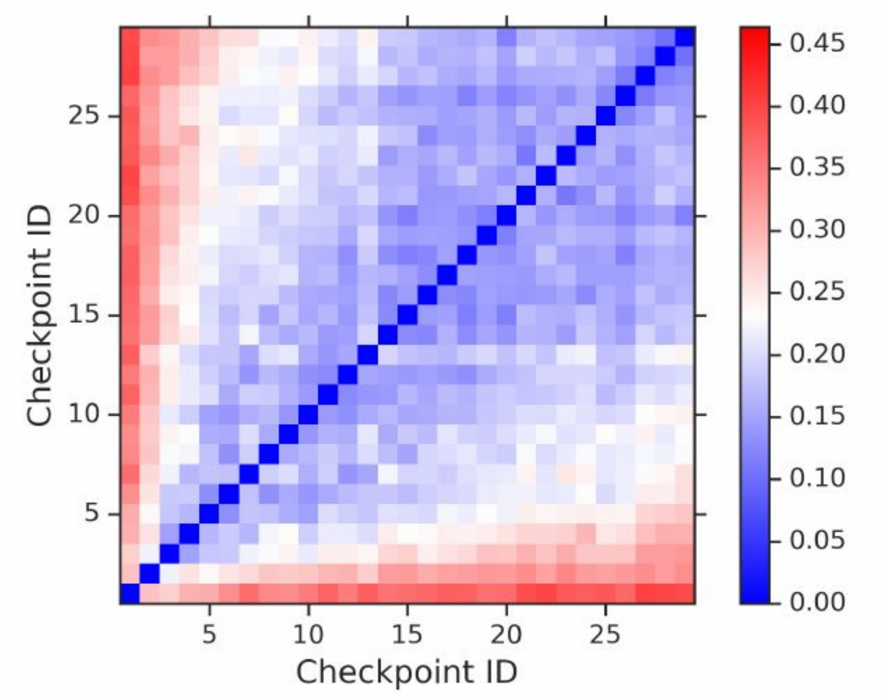
$$\frac{1}{N}\sum_{n=1}^{N}\left(f(x_n;\vec{w}_1) \neq f(x_n;\vec{w}_2)\right)$$

Function space distance = prediction dissimilarity

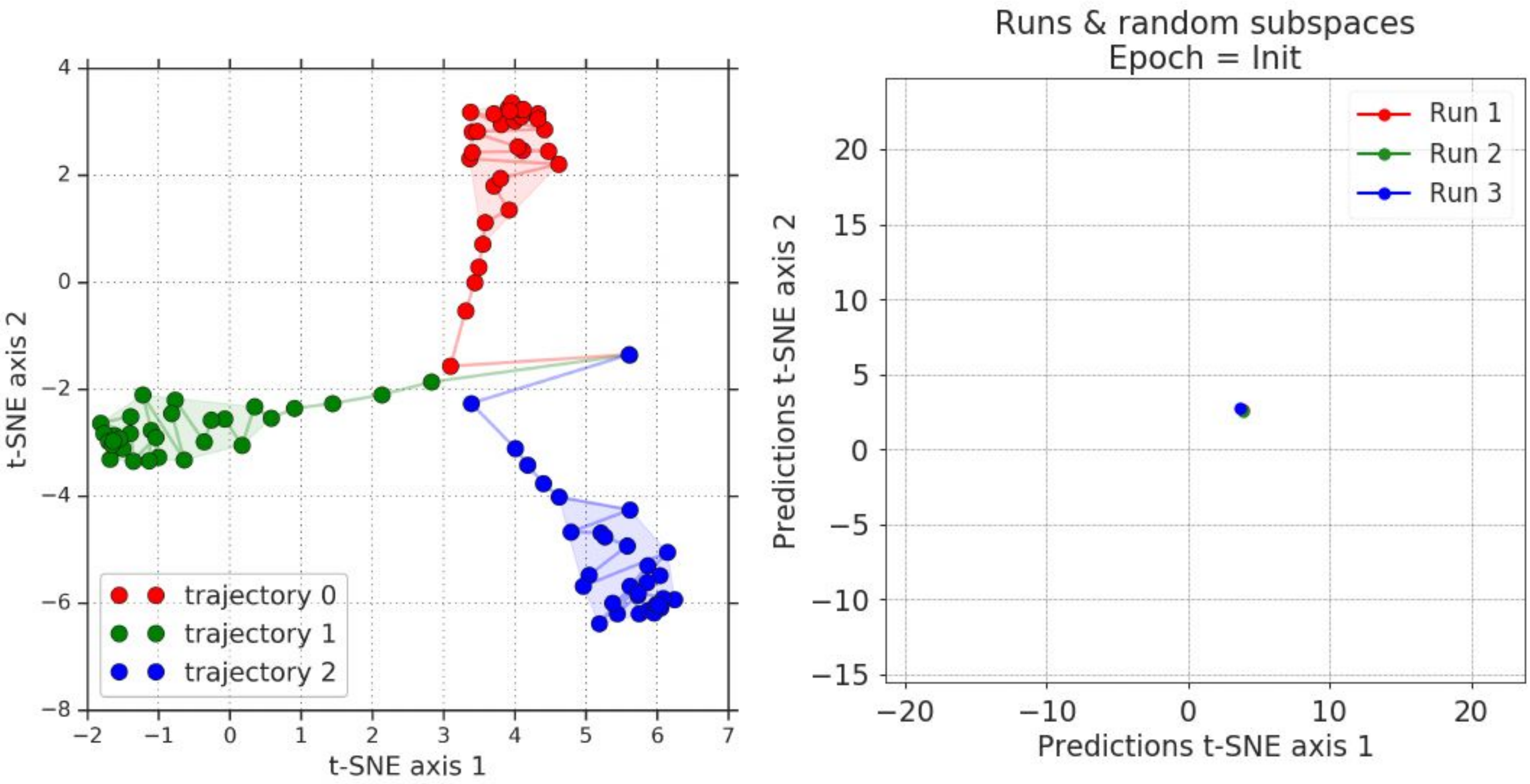
Equally dissimilar between runs



Similar to itself within run

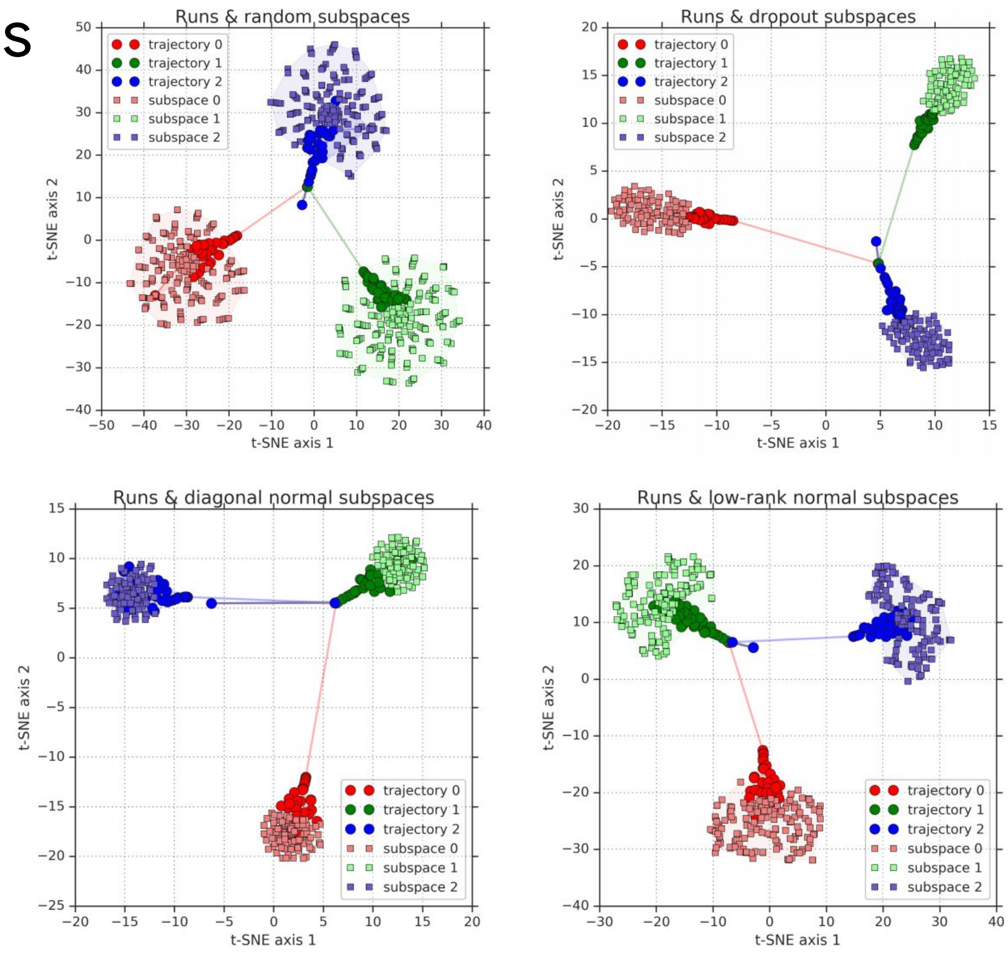


Predictions similarity within and across trajectories

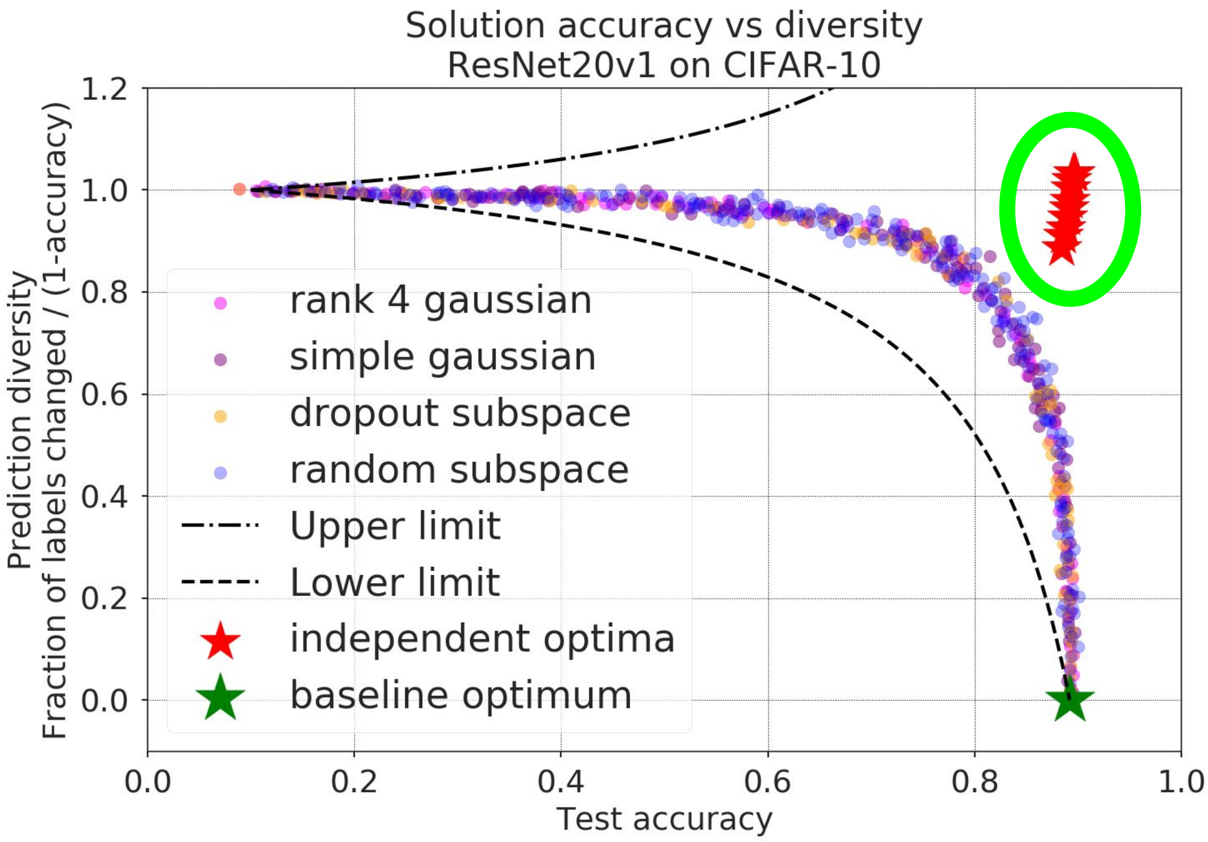


Subspace sampling methods

- Random
- Diagonal Gaussian
- Low-rank Gaussian
- Monte Carlo Dropout



Prediction diversity vs Accuracy



- From a bias-variance tradeoff perspective, we care about low bias as well as diversity.
- We plot fraction of data points where top-1 prediction is different. Note that maximum disagreement depends on the accuracy (completely different mistakes).
- Function diversity is higher across trajectories than within trajectories.

Loss landscape and function space similarity of 2 trajectories

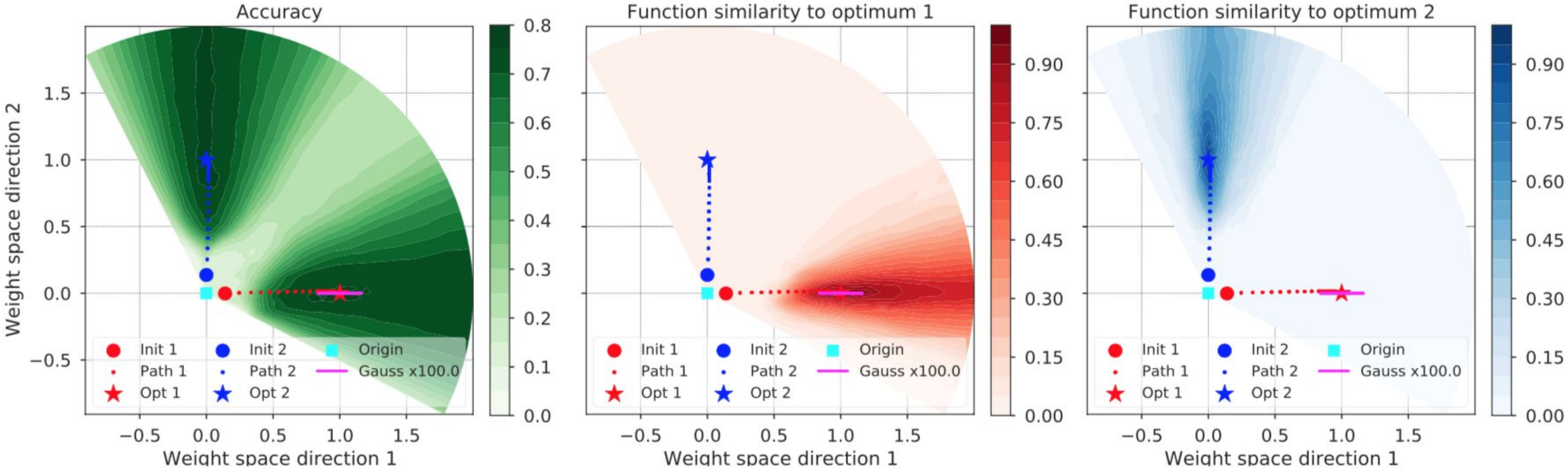
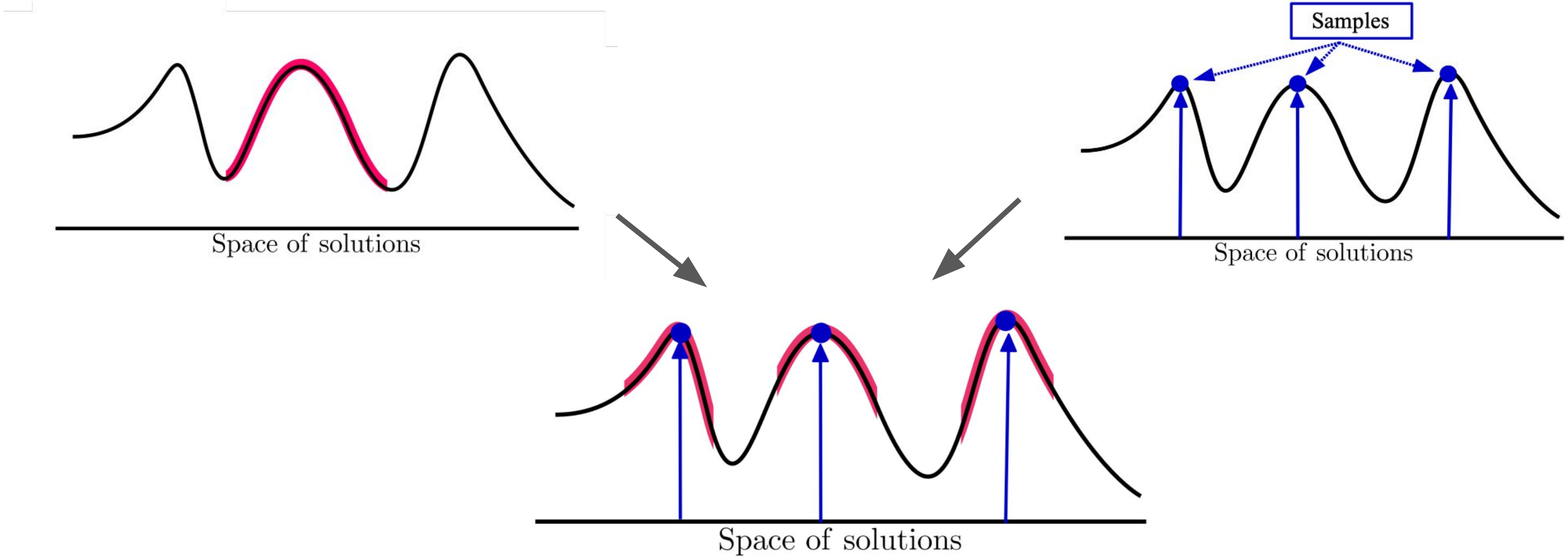


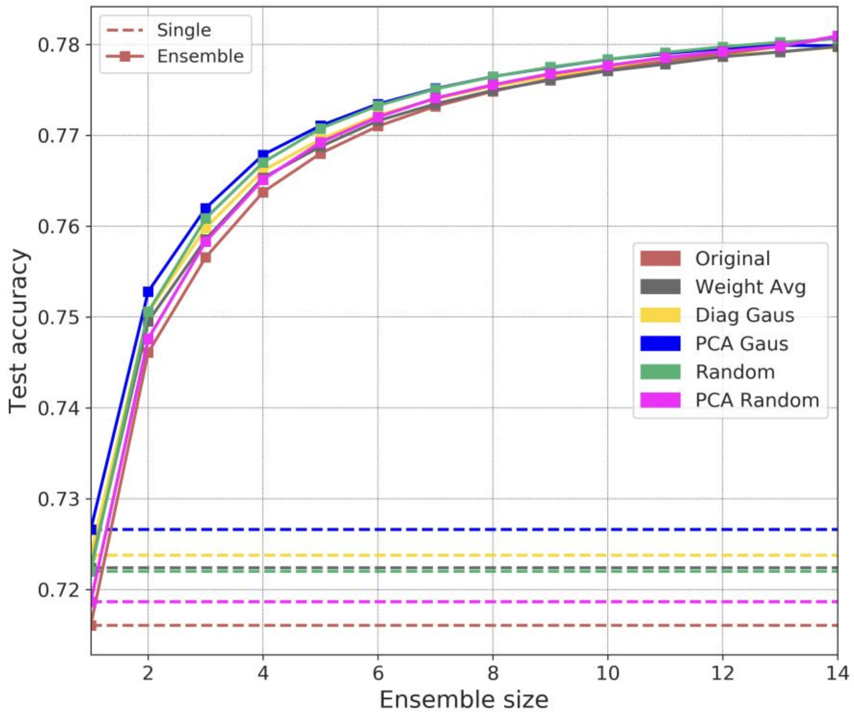
Figure 5: *Results using MediumCNN on CIFAR-10*: Radial loss landscape cut between the origin and two independent optima. Left plot shows accuracy of models along the paths of the two independent trajectories, and the middle and right plots show function space similarity to the two optima.

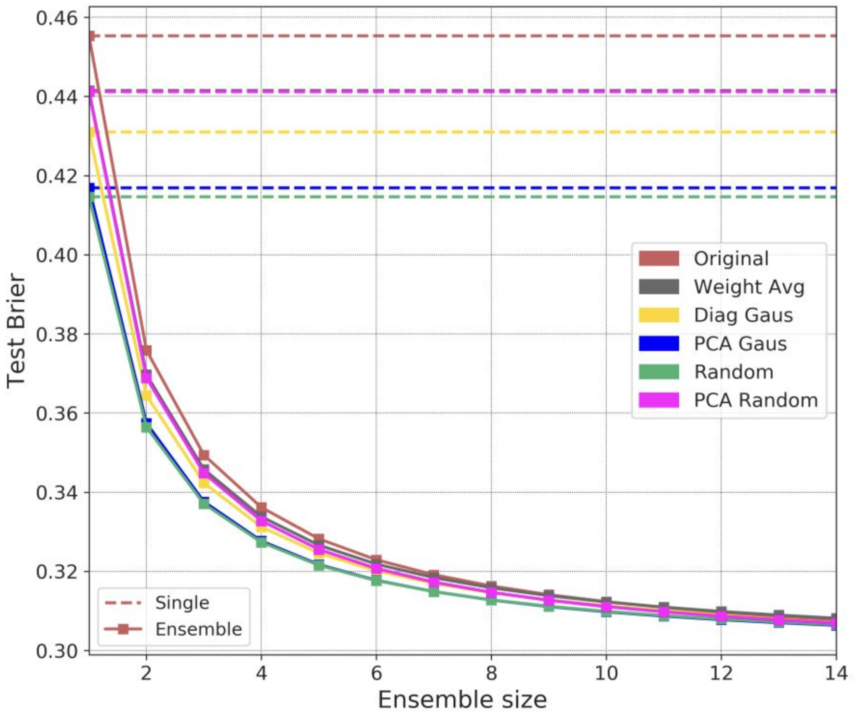
Can we combine uncertainty from subspace (within-mode) and ensembles (across-mode)?



Idea: run multiple trajectories and use subspace sampling within each trajectory (e.g. Ensemble + Diagonal Gaussian)

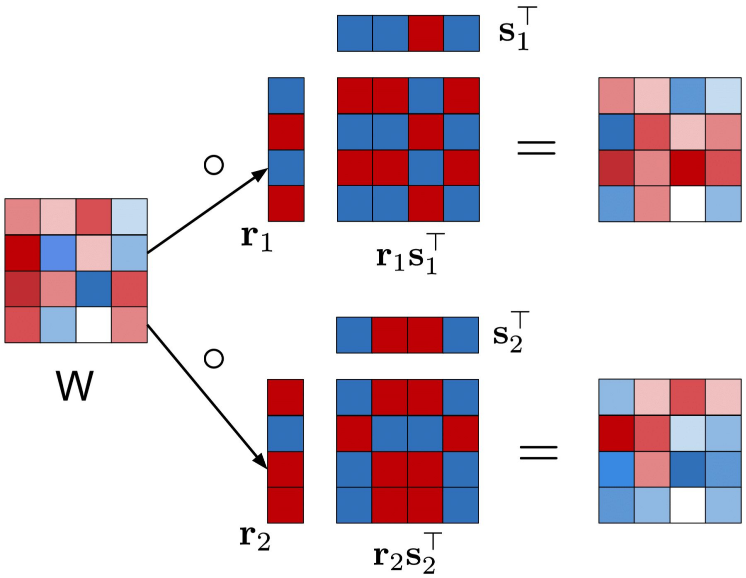
Best of both worlds - Results on CIFAR-10 using MediumCNN





Follow up

Efficient Ensembles and BNNs by Sharing Parameters



Parameterize each weight matrix as a new weight matrix ***W*** multiplied by the outer product of two vectors ***r*** and **s**.

There is an independent set of ***r*** and **s** vectors for each ensemble member; ***W*** is shared.

Known as **BatchEnsemble or Rank-1 Ensemble.**

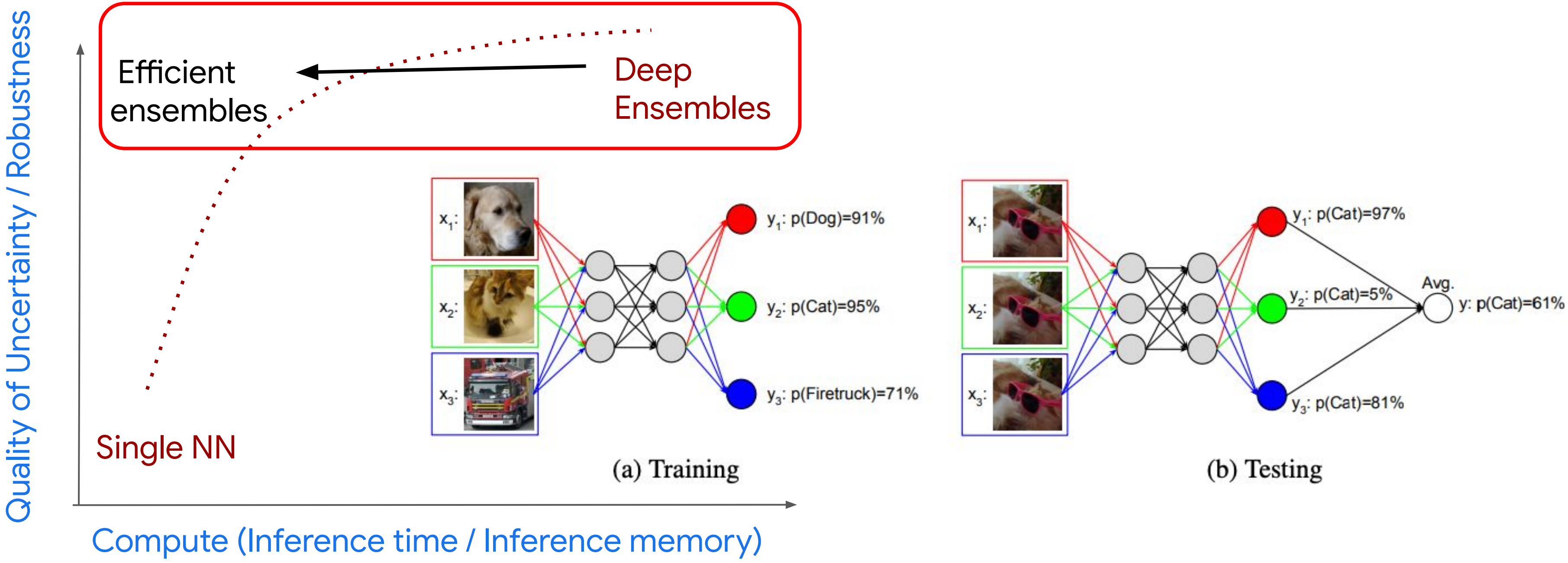
Can also construct **Rank-1 Bayesian NNs.**

[Wen+ 2020, Dusenberry+ 2020]

Efficient ensembles lower inference memory and inference time

Google AI
Brain Team

Idea: Create multiple input heads + multiple output heads (MIMO). Train each head on different SGD batch.



[Training Independent Subnetworks for Robust Prediction. Havasi+ 2020]

Bayesian Deep Ensembles via the Neural Tangent Kernel

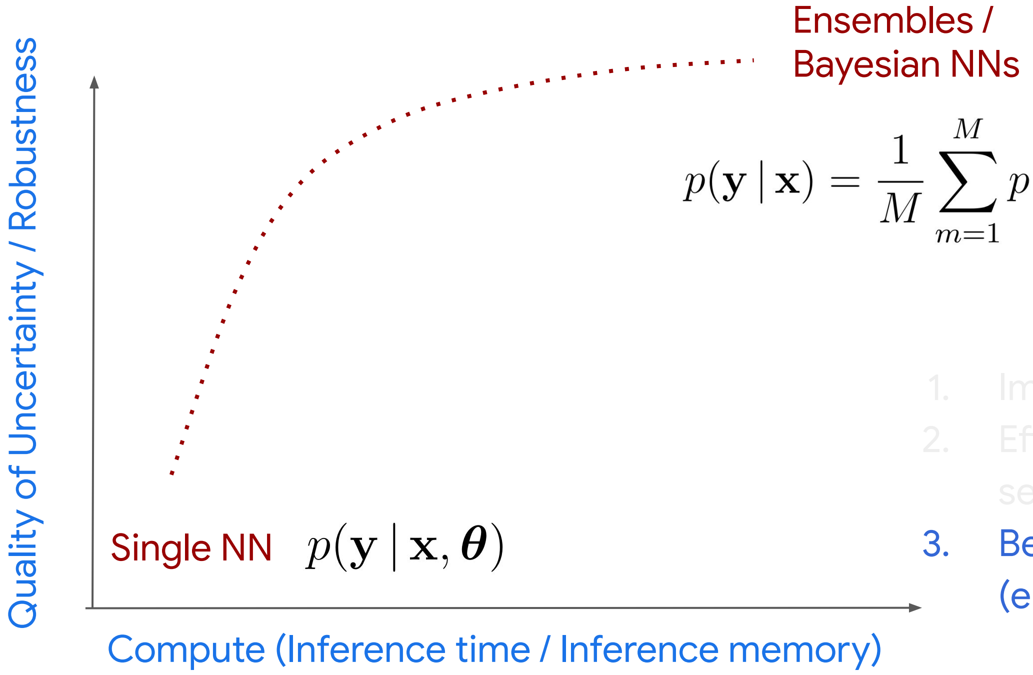
Bayesian Deep Ensembles recover the correct posterior in the infinite width limit.
Deep Ensembles approximate the posterior mean well.

Table 1: Predictive distributions of wide ensembles for various training methods. *std* denotes standard training with $f(\boldsymbol{x}, \boldsymbol{\theta})$, and *ours* denotes training using our additive $\delta(\boldsymbol{x})$ to make $\tilde{f}(\boldsymbol{x}, \boldsymbol{\theta})$.

Method	Layers trained	Output Noise	$\mu(\boldsymbol{x})$	$\Sigma(\boldsymbol{x}, \boldsymbol{x}')$
NNGP	Final	$\sigma^2 \geq 0$	$\mathcal{K}_{\boldsymbol{x}\mathcal{X}}(\mathcal{K}_{\mathcal{X}\mathcal{X}}+\sigma^2 I)^{-1}\mathcal{Y}$	$\mathcal{K}_{\boldsymbol{x}\boldsymbol{x}'} - \mathcal{K}_{\boldsymbol{x}\mathcal{X}}(\mathcal{K}_{\mathcal{X}\mathcal{X}}+\sigma^2 I)^{-1}\mathcal{K}_{\mathcal{X}\boldsymbol{x}'}$
Deep Ensembles	All (*std*)	$\sigma^2 = 0$	$\Theta_{\boldsymbol{x}\mathcal{X}}\Theta_{\mathcal{X}\mathcal{X}}^{-1}\mathcal{Y}$	$\mathcal{K}_{\boldsymbol{x}\boldsymbol{x}'} - (\Theta_{\boldsymbol{x}\mathcal{X}}\Theta_{\mathcal{X}\mathcal{X}}^{-1}\mathcal{K}_{\mathcal{X}\boldsymbol{x}'} + h.c.)$ $\Theta_{\boldsymbol{x}\mathcal{X}}\Theta_{\mathcal{X}\mathcal{X}}^{-1}\mathcal{K}_{\mathcal{X}\mathcal{X}}\Theta_{\mathcal{X}\mathcal{X}}^{-1}\Theta_{\mathcal{X}\boldsymbol{x}'}$
Randomised Prior	All (*std*)	$\sigma^2 > 0$	$\Theta_{\boldsymbol{x}\mathcal{X}}(\Theta_{\mathcal{X}\mathcal{X}}+\sigma^2 I)^{-1}\mathcal{Y}$	$\mathcal{K}_{\boldsymbol{x}\boldsymbol{x}'} - (\Theta_{\boldsymbol{x}\mathcal{X}}(\Theta_{\mathcal{X}\mathcal{X}}+\sigma^2 I)^{-1}\mathcal{K}_{\mathcal{X}\boldsymbol{x}'} + h.c.)$ $+\Theta_{\boldsymbol{x}\mathcal{X}}(\Theta_{\mathcal{X}\mathcal{X}}+\sigma^2 I)^{-1}(\mathcal{K}_{\mathcal{X}\mathcal{X}}+\sigma^2 I)(\Theta_{\mathcal{X}\mathcal{X}}+\sigma^2 I)^{-1}\Theta_{\mathcal{X}\boldsymbol{x}'}$
NTKGP	All (*ours*)	$\sigma^2 \geq 0$	$\Theta_{\boldsymbol{x}\mathcal{X}}(\Theta_{\mathcal{X}\mathcal{X}}+\sigma^2 I)^{-1}\mathcal{Y}$	$\Theta_{\boldsymbol{x}\boldsymbol{x}'} - \Theta_{\boldsymbol{x}\mathcal{X}}(\Theta_{\mathcal{X}\mathcal{X}}+\sigma^2 I)^{-1}\Theta_{\mathcal{X}\boldsymbol{x}'}$

[Bayesian Deep Ensembles via the Neural Tangent Kernel. He+ 2020]

Better inductive biases for representations



Single NN $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$

Ensembles / Bayesian NNs

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_m)$$

“Composable” toolkit

1. Improve the “base” model p(y|x,θ)
2. Efficiently average predictions over diverse set of functions θ_1, θ_2 ... θ_M
3. Better inductive biases for representations (e.g. pre-training or data augmentation)

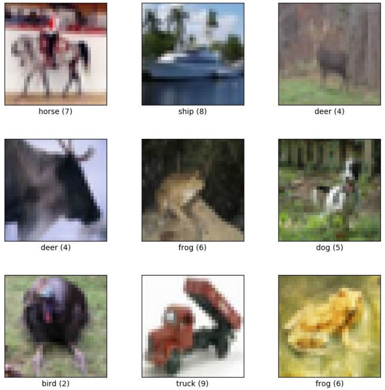
Exploring the limits of OOD detection

Stanislav Fort*, Jie Ren*, Balaji Lakshminarayanan

Google AI
Brain Team

Goal: Improve SOTA on hard OOD detection tasks

Far OOD, AUROC = 99%

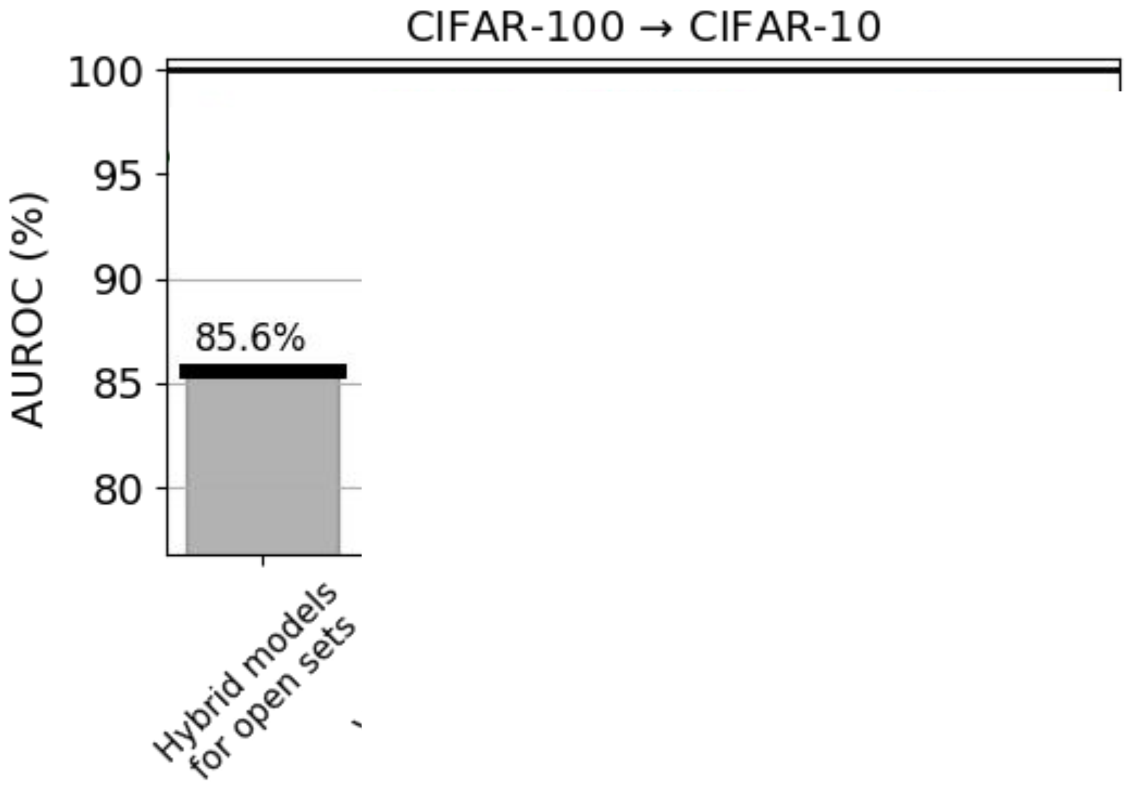


CIFAR-10 (ID) **SVHN (OOD)**

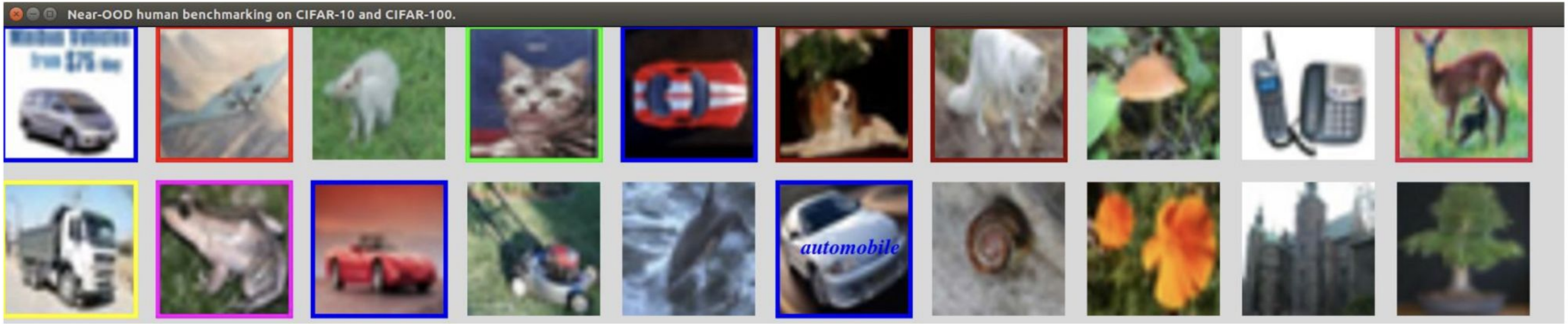
Near OOD, AUROC = 85%

CIFAR-100 (ID) **CIFAR-10 (OOD)**

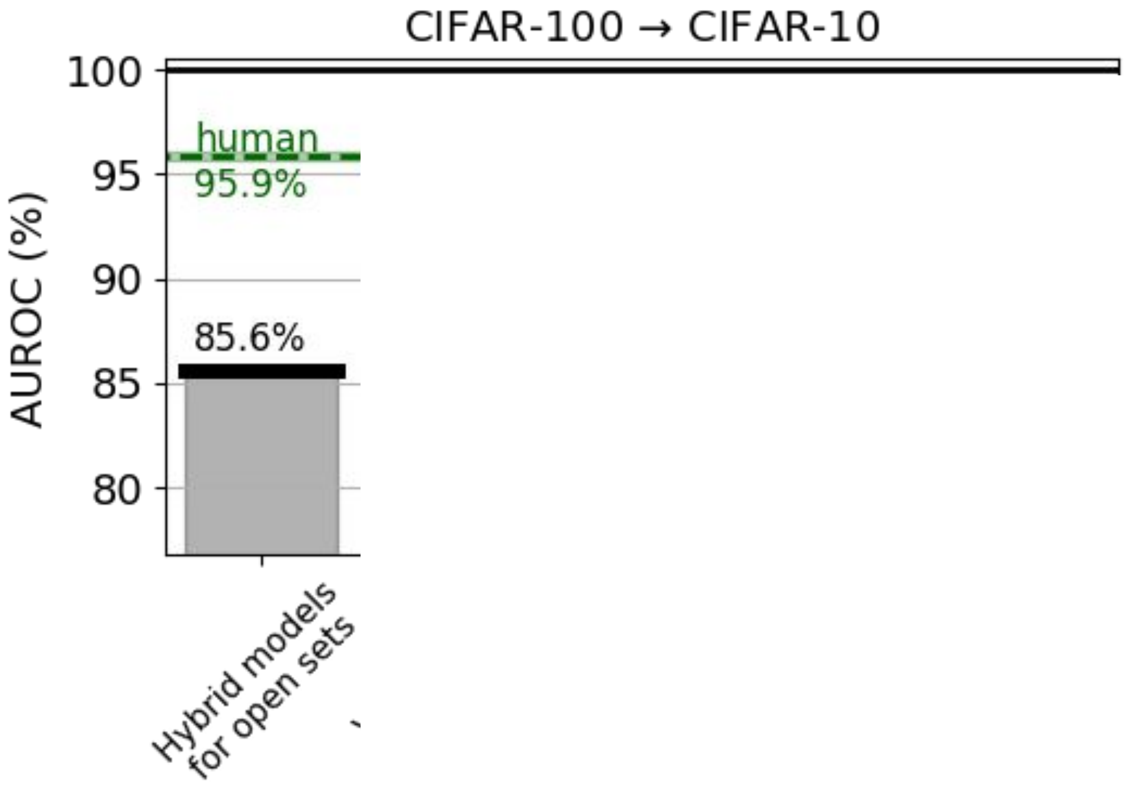
Improving near-OOD detection



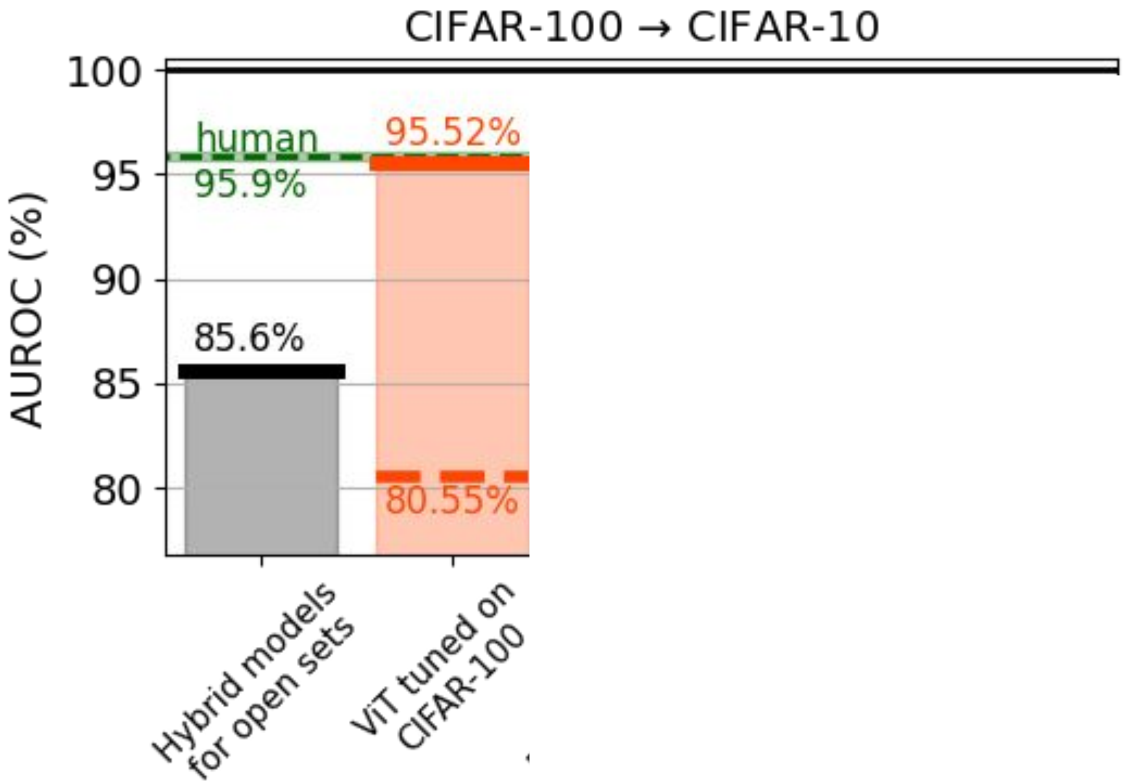
Benchmarking Human Performance (S.F. is the human)



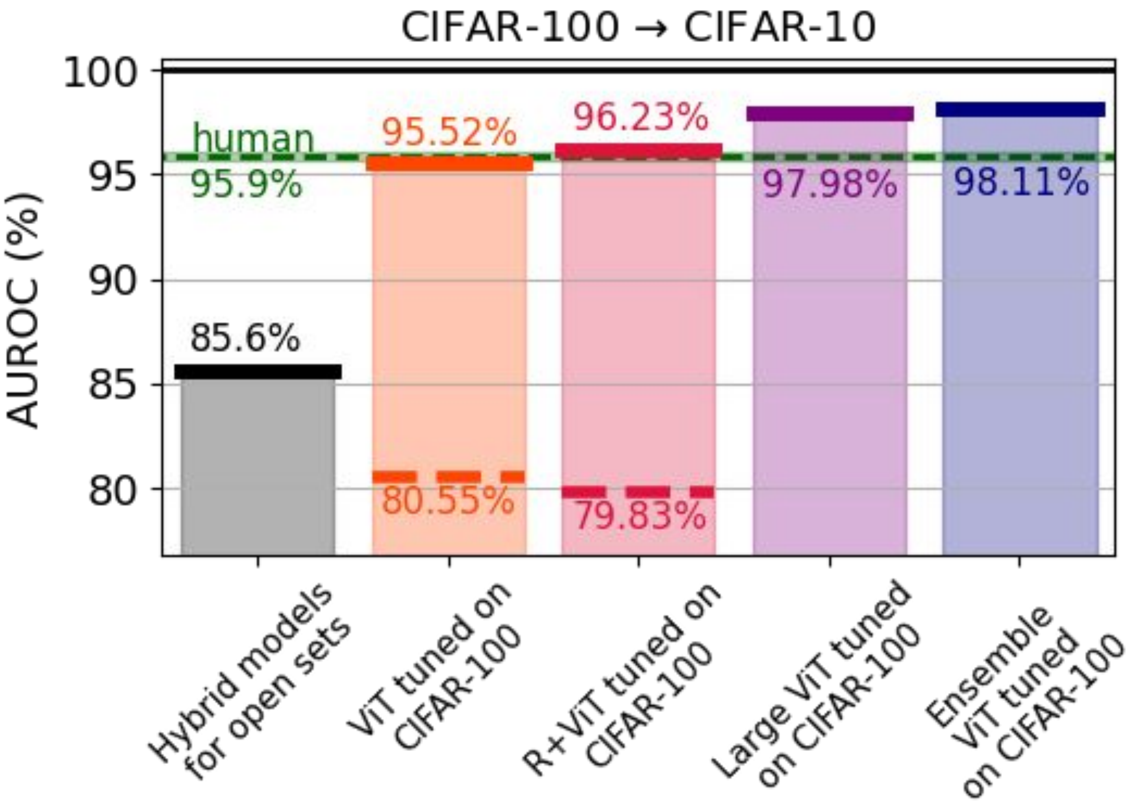
Improving near-OOD detection



Pre-trained Vision Transformers improve near-OOD detection



Pre-trained Vision Transformers improve near-OOD detection



Pre-trained ViT improves near-OOD detection

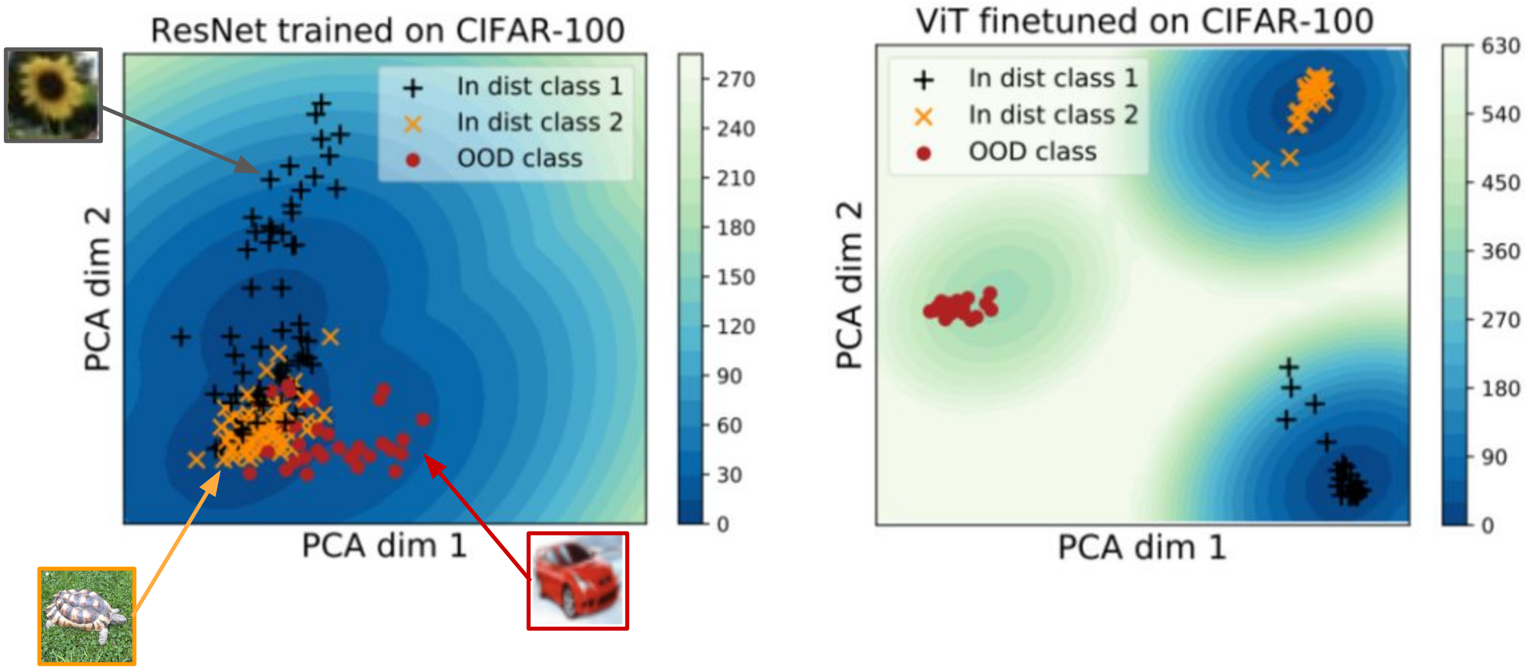
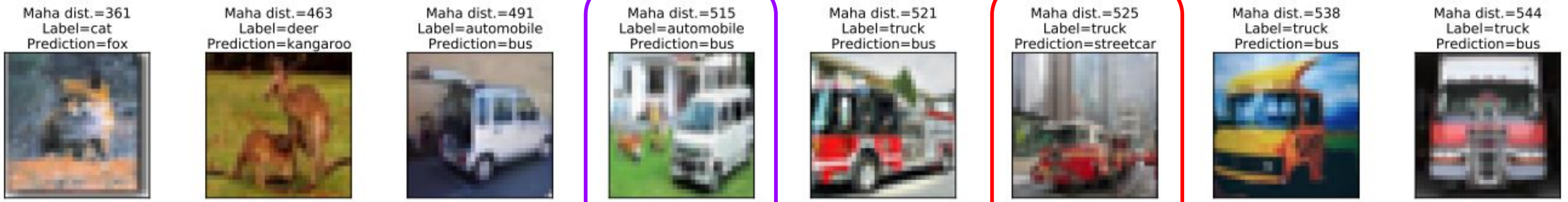


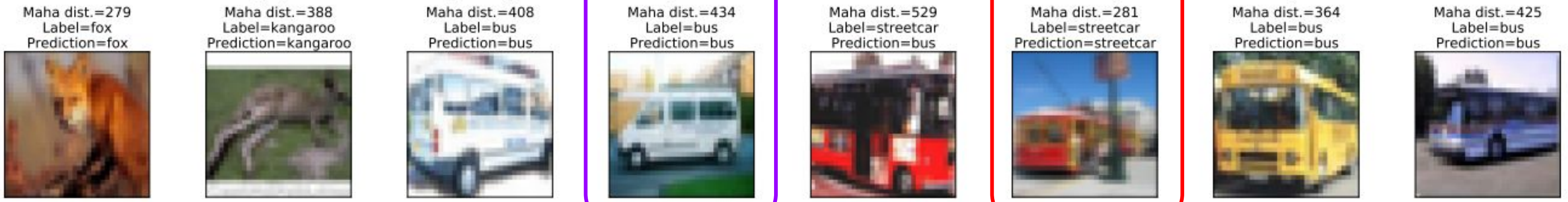
Figure: 2D PCA project of the space of embedding.
Color coding shows Mahalanobis outlier score.

Qualitative failure cases of ViT OOD detection

- Most false positives are due to mislabeling or ambiguity



(a) OOD images (CIFAR-10) closest to the in-distribution (CIFAR-100).

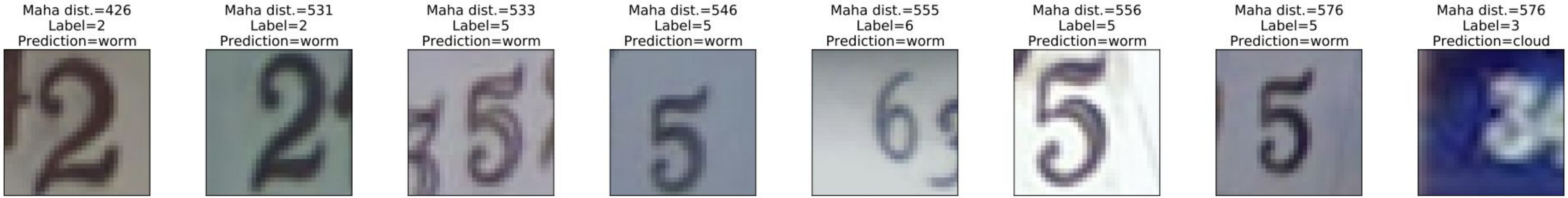


(b) The in-distribution (CIFAR-100) images with the closest embedding vector to images in Figure 11a.

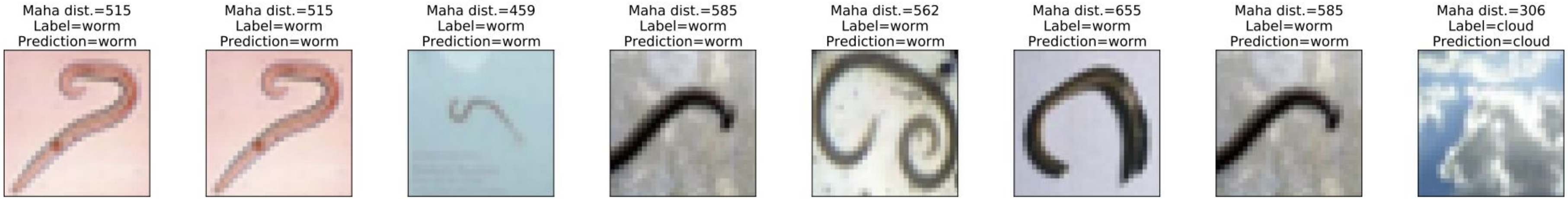
Google AI
Brain Team

Qualitative failure cases of ViT OOD detection

- SVHN digits are classified as CIFAR-100 “worms”

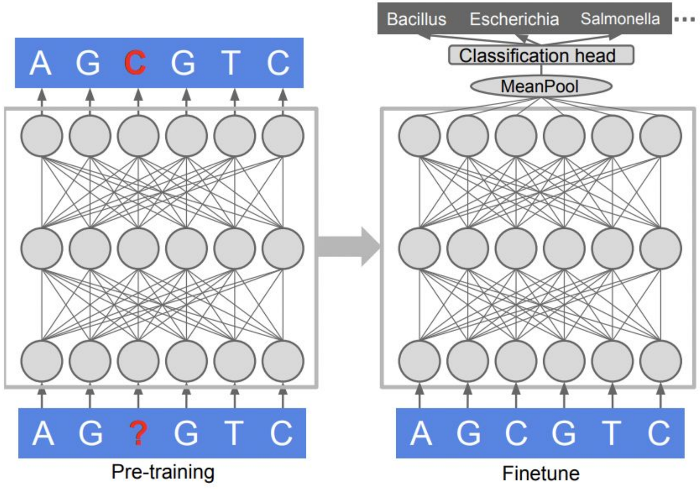


(a) OOD images (SVHN) closest to the in-distribution (CIFAR-100).



(b) The in-distribution (CIFAR-100) images with the closest embedding vector to images in Figure 11a.

The same story for genomics
(here pre-trained in a self-supervised way)



(a) BERT pre-training and fine-tuning for genomics.

Model	Test Accuracy	Mahalanobis AUROC	MSP AUROC
1D CNN [Ren et al., 2019]	85.93±0.11%	64.75±0.73%	65.84±0.46%
BERT pre-train and fine-tune	89.84±0.00%	**77.49±0.04%**	73.53±0.03%

Putting it all together

Simple, Composable Recipe to Improve “Out-of-the-box” Reliability

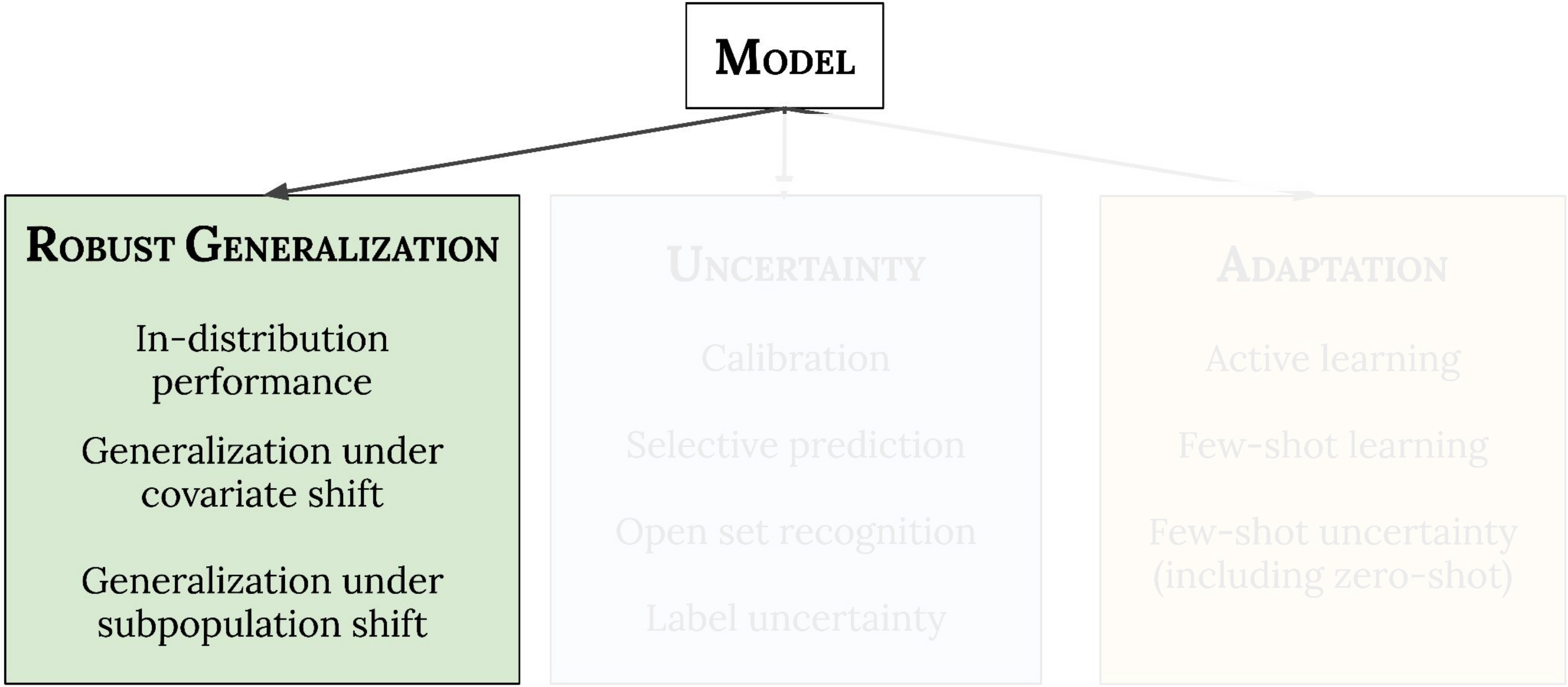
Plex: Towards Reliability Using Pretrained Large Model Extensions

Dustin Tran[*1], **Jeremiah Liu**[1], **Michael W. Dusenberry**[1], **Du Phan**[1], **Mark Collier**[1], **Jie Ren**[1], **Kehang Han**[1], **Zi Wang**[1], **Zelda Mariet**[1], **Huiyi Hu**[1], **Neil Band**[2], **Tim G. J. Rudner**[2], **Karan Singhal**[1], **Zachary Nado**[1], **Joost van Amersfoort**[2], **Andreas Kirsch**[2], **Rodolphe Jenatton**[1], **Nithum Thain**[1], **Honglin Yuan**[1†], **Kelly Buchanan**[1†], **Kevin Murphy**[1], **D. Sculley**[1], **Yarin Gal**[2], **Zoubin Ghahramani**[1], **Jasper Snoek**[1], **Balaji Lakshminarayanan**[1]

[1]*Google* [2]*University of Oxford*

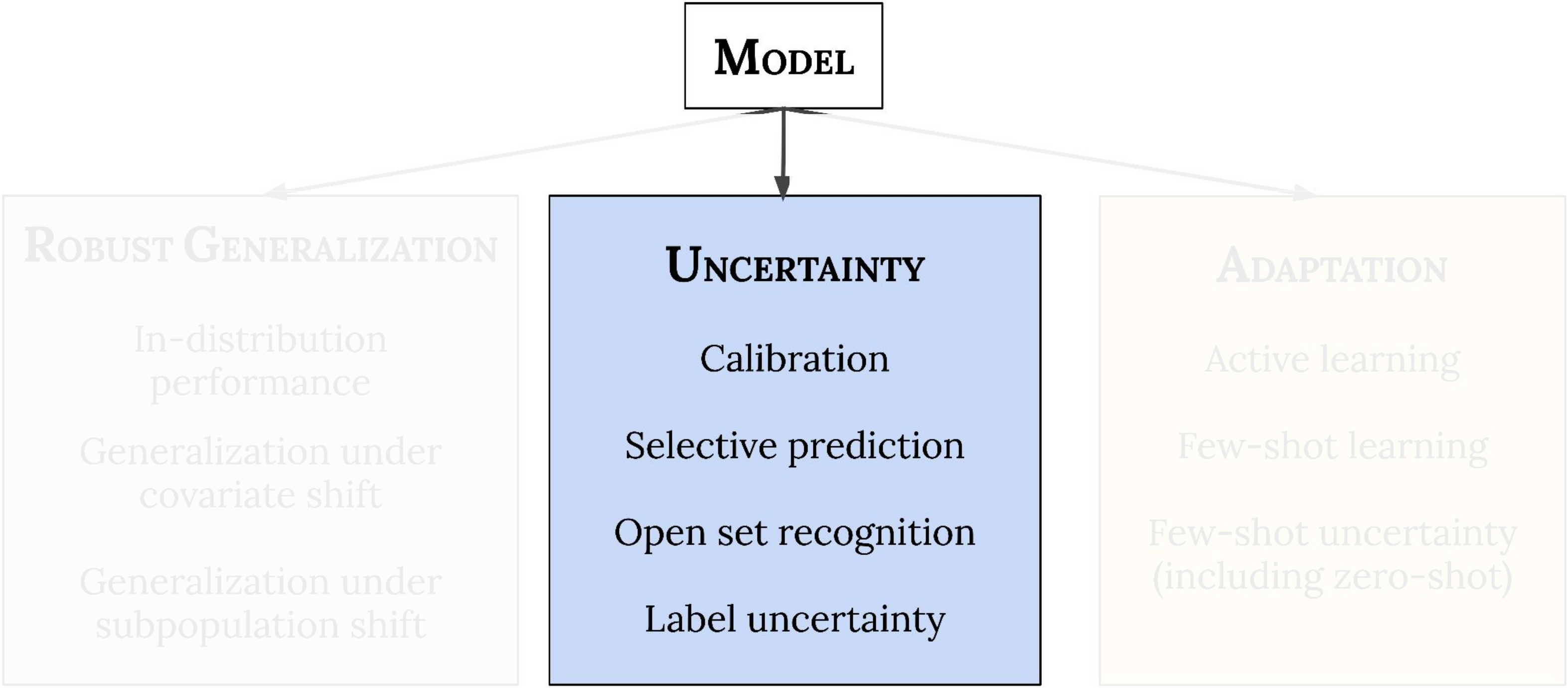
Paper: https://goo.gle/plex-paper
Code: https://goo.gle/plex-code
Blog: https://ai.googleblog.com/2022/07/towards-reliability-in-deep-learning.html

Reliability as a Goal for AI



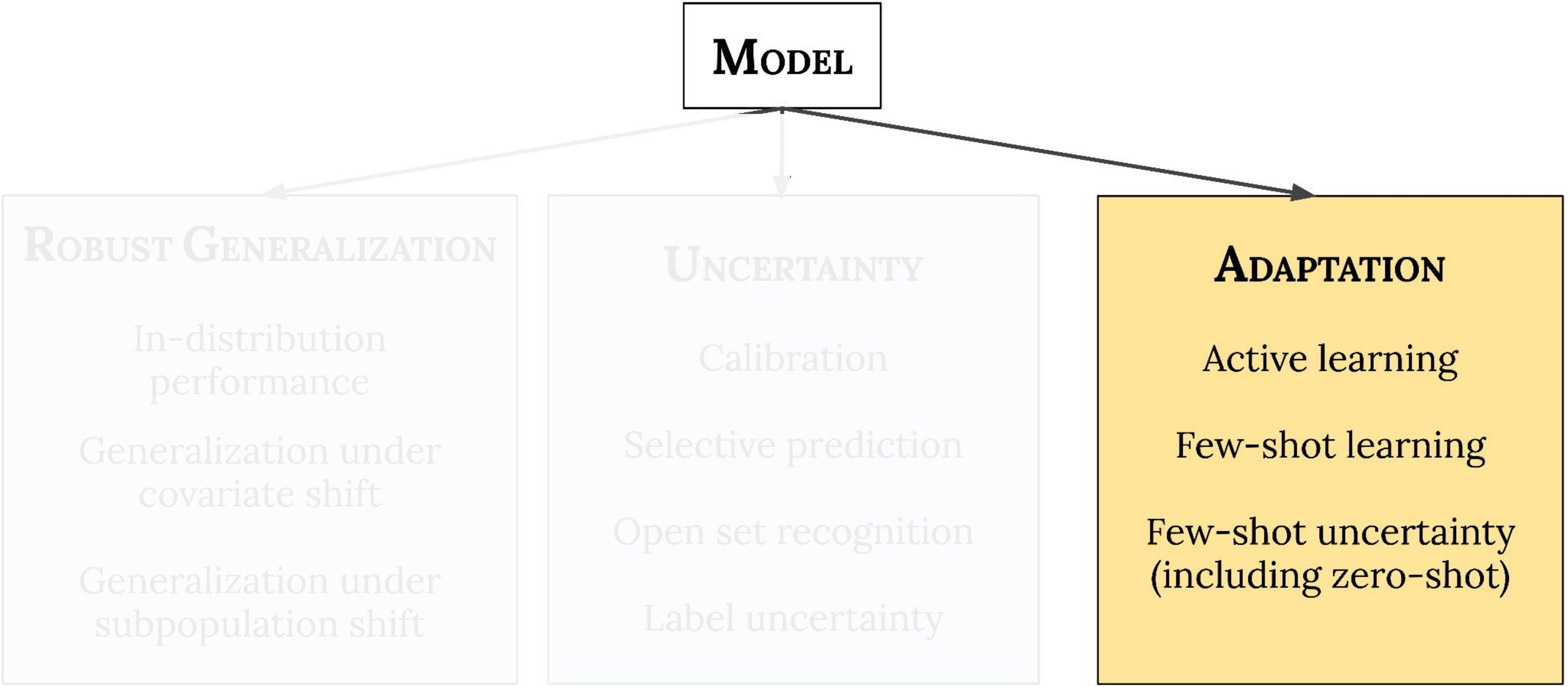
Predict well across
distribution shifts

Reliability as a Goal for AI



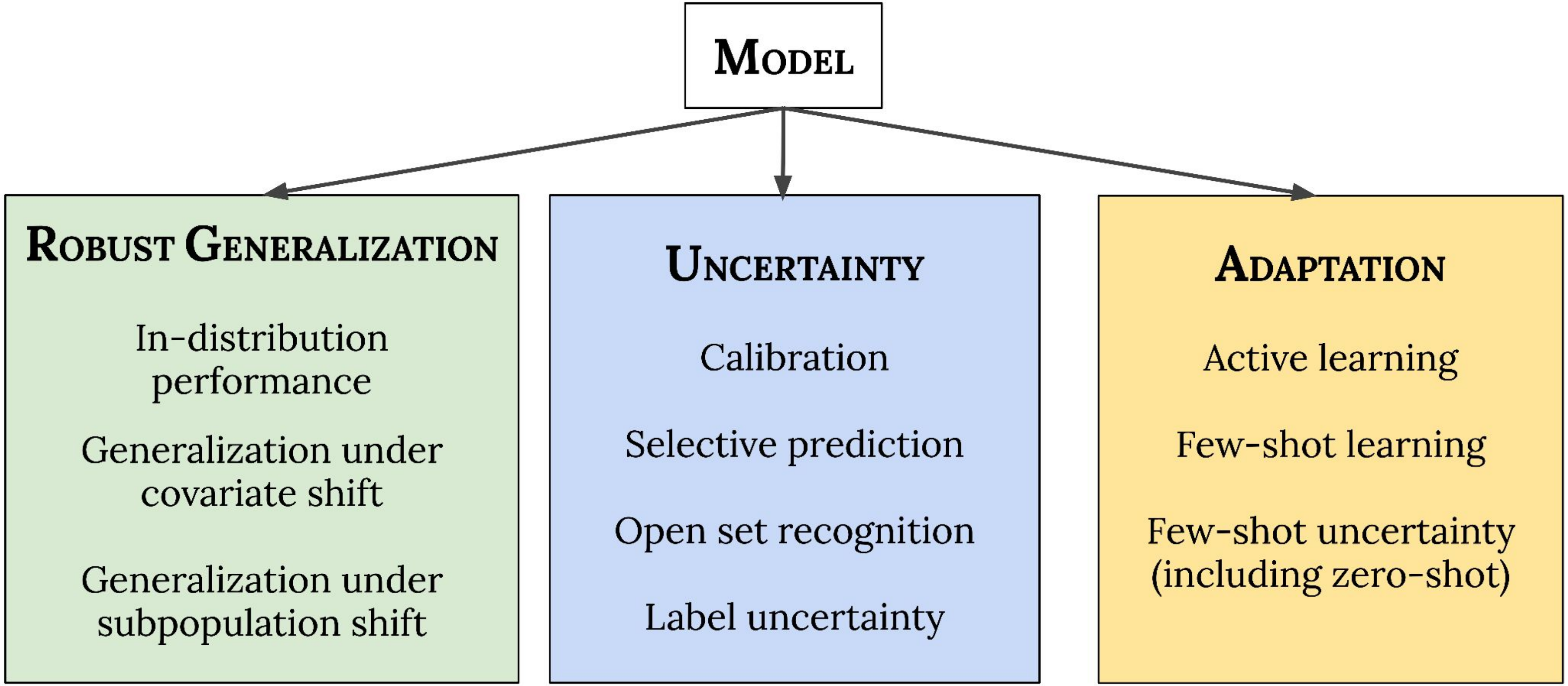
Know what they don't know

Reliability as a Goal for AI



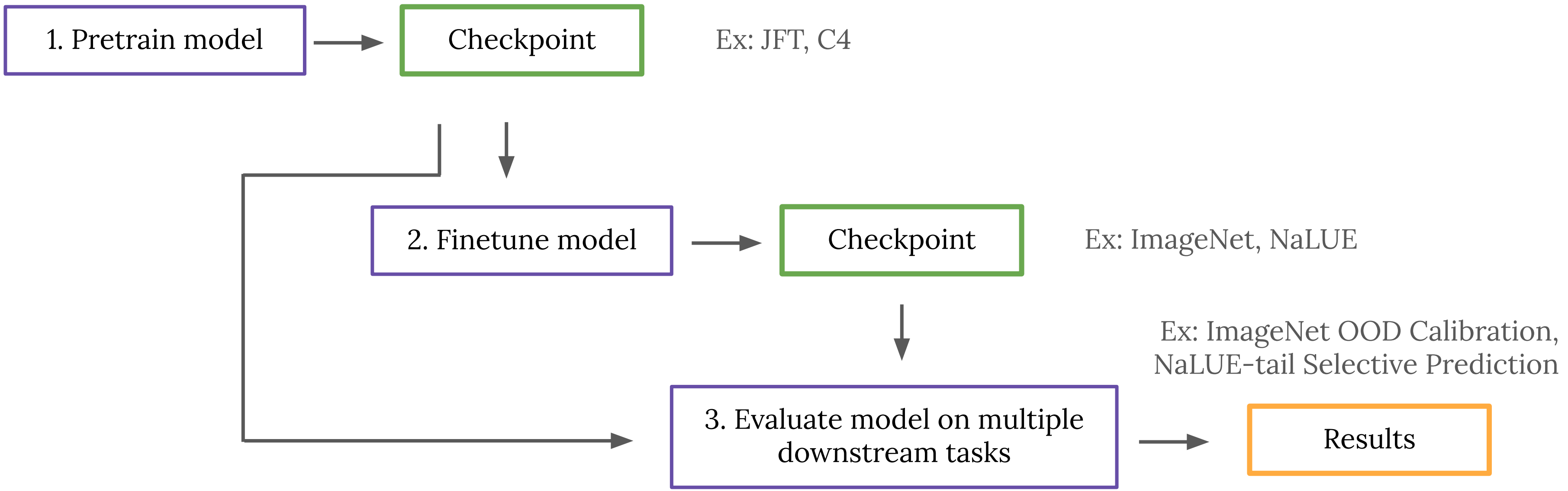
Learn what they don't know quickly / efficiently

Reliability as a Goal for AI



Setting

[goo.gle/plex-paper]



[goo.gle/plex-paper]

Vision Datasets & Tasks

Pretraining

- JFT-300M
  - 300M images, 375M labels (multi-label), 18291 classes, 20% of labels are noisy
- JFT-4B
  - 4B images, 6.7B labels (multi-label), 29592 classes
- ImageNet21k
  - 13M images, 14M labels (multi-label), 21843 classes, super-set of ImageNet

Finetuning

- ImageNet
- CIFAR-10/100
- RETINA (diabetic retinopathy)
- Places365

Evaluate "Out-of-the-box" performance

- ImageNet-{C,A,R,V2,Vid-robust}, YTBB-robust
  - Covariate shifts from ImageNet
- RETINA Country Shift // RETINA Severity Shift
  - Covariate shift // OOD from RETINA
- ImageNet-ReaL-H // CIFAR-10-H
  - Label uncertainty from ImageNet // CIFAR-10
- CIFAR-10, SVHN // CIFAR-100, SVHN
  - OOD from CIFAR-10 // CIFAR-100
- SP-CIFAR-10 // SP-CIFAR-100
  - Subpopulation shift from CIFAR-10 // CIFAR-100
- 7 smaller datasets for few-shot adaptation

[goo.gle/plex-paper]

Language Datasets & Tasks

Pretraining

- C4 (Colossal Clean Crawled Corpus)
  - 355M webpages (~7 TB) from Common Crawl scrapes for unsupervised masked language modeling (MLM) pretraining.

Tasks

- Natural Language Inference (NLI).
  - Predict whether sentence1 entails sentence2.
  - "*A boy crying; The boy is not happy.*" ➞ "`entail`".
- Toxic Comment Detection.
  - Predict toxicity of a sentence.
  - "*I'm gay and I'm proud.*" ➞ "`non-toxic`"
- Natural Language Understanding (NLU).
  - Predict `Vertical`, `Domain`, `Intent` of a user query.
  - "*Turn on radio*" ➞ "`Vertical=Media; Domain=Radio; Intent=TurnOn`".

Evaluation

	In-domain Generalization	Out-of-Domain Generalization	Subpopulation Shift
Natural Language Inference	MultiNLI, Matched split. Sentences from multiple genres.	MultiNLI, mismatched split Sentences from genres different from *matched*.	HANS Adversarial examples attack heuristics that neural models rely on.
Toxic Comment Detection	WikipediaTalk 200K Wikipedia editor conversations.	CivilComments 2M news website comments 2015-2017	CivilComments-Identity Subset of CivilComments with socio-ethnic identity mentions
NLU	NaLUE. Chatbot queries from 18 verticals, 77 domains, and ~260 intents	NaLUE, Out-of-scope Set Out-of-scope queries never appeared in training set.	NaLUE, Tail Intents Subset of tail intents from NaLUE

[goo.gle/plex-paper]

Base models

Vision: **Vision Transformer**

- Large (L/32)
  - ~325M parameters
- Base (B/32)
  - ~87M parameters
- Small (S/32)
  - ~22M parameters

[Dosovitskiy+ 2021]

Text: **T5 1.1**

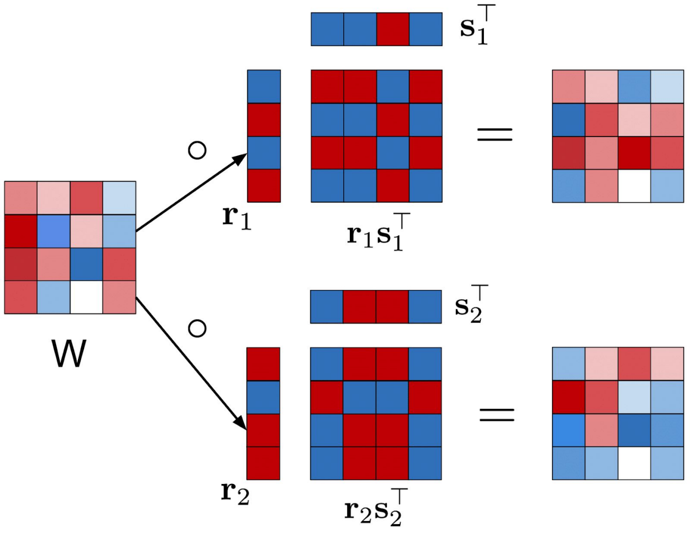
- Large
  - ~880M parameters
- Base
  - ~250M parameters
- Small
  - ~77M parameters

[Raffel+ 2020]

Plex's key ingredients

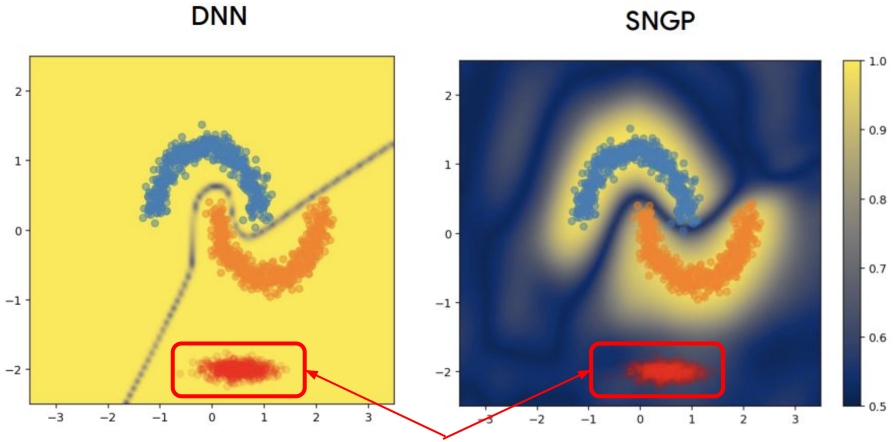
[goo.gle/plex-paper]

- **Massive pre-training.** Pretrain on JFT with 300M to 4B images; and C4 for text.
- **Base Transformer architecture.** ViT for vision, T5 for text.



Efficient ensembles with **BatchEnsemble.**

[Wen+ 2020]



Last layer changes

Gaussian process layer from **SNGP** (reduces confidence far away from training data).

Heteroscedastic layer (aleatoric label uncertainty).

[goo.gle/plex-paper]

Final Plex models

Vision: **ViT-Plex**

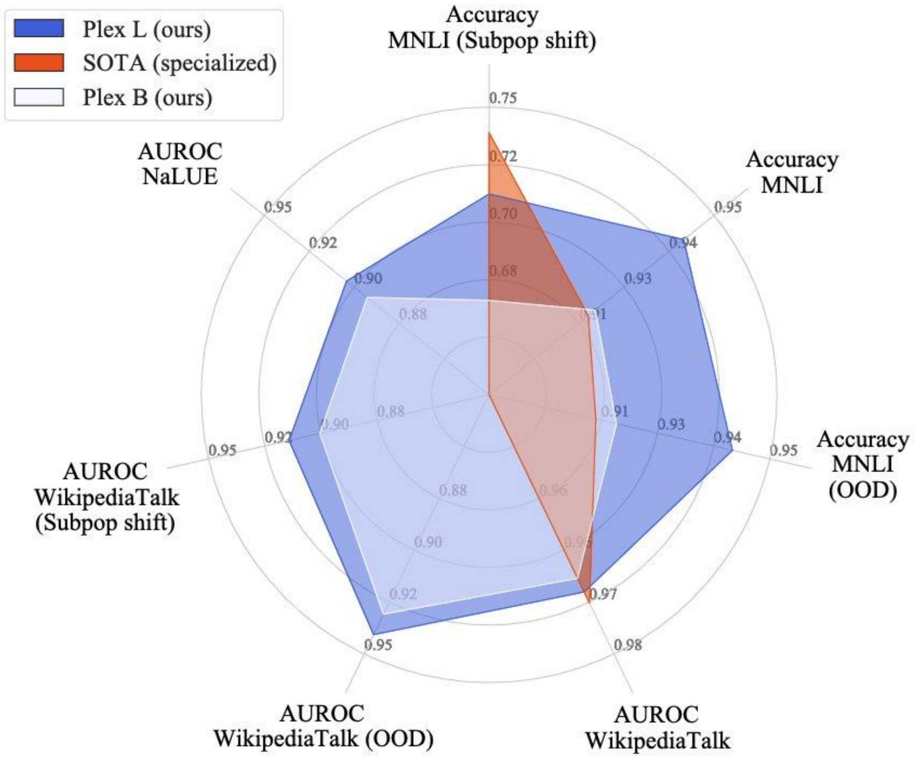
- **Vision Transformer** (ViT) **Large**
- Pretrained with **BatchEnsemble** (final few layers).
- Finetuned with **BatchEnsemble** (still final few layers) **+ Heteroscedastic** last layer.

Text: **T5-Plex**

- **T5 1.1 Large**
- Uses existing pretrained checkpoint.
- Finetuned with **BatchEnsemble** (final few layers) **+ GP** last layer.

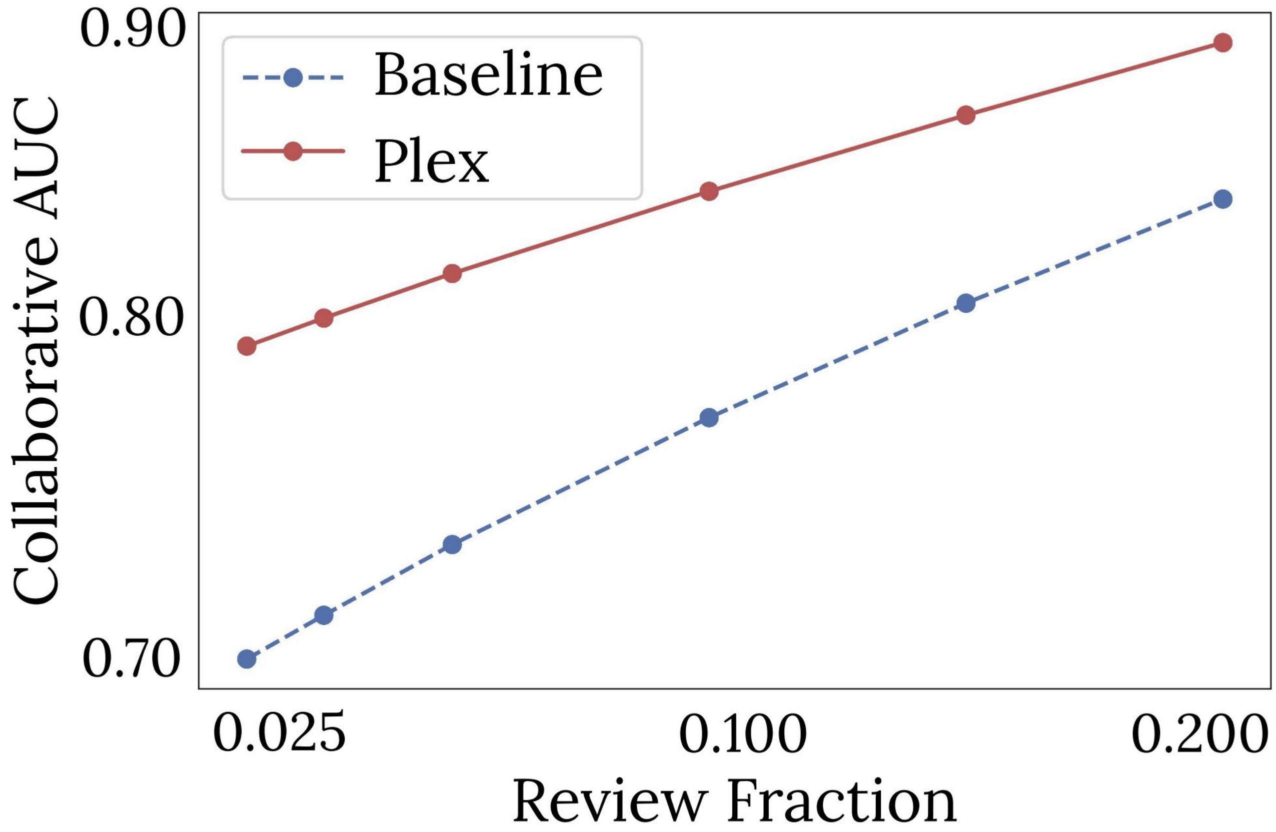
Results: ViT-Plex and T5-Plex

[goo.gle/plex-paper]



Highlights: Selective Prediction

[goo.gle/plex-paper]



Highlights: Structured Open Set Recognition

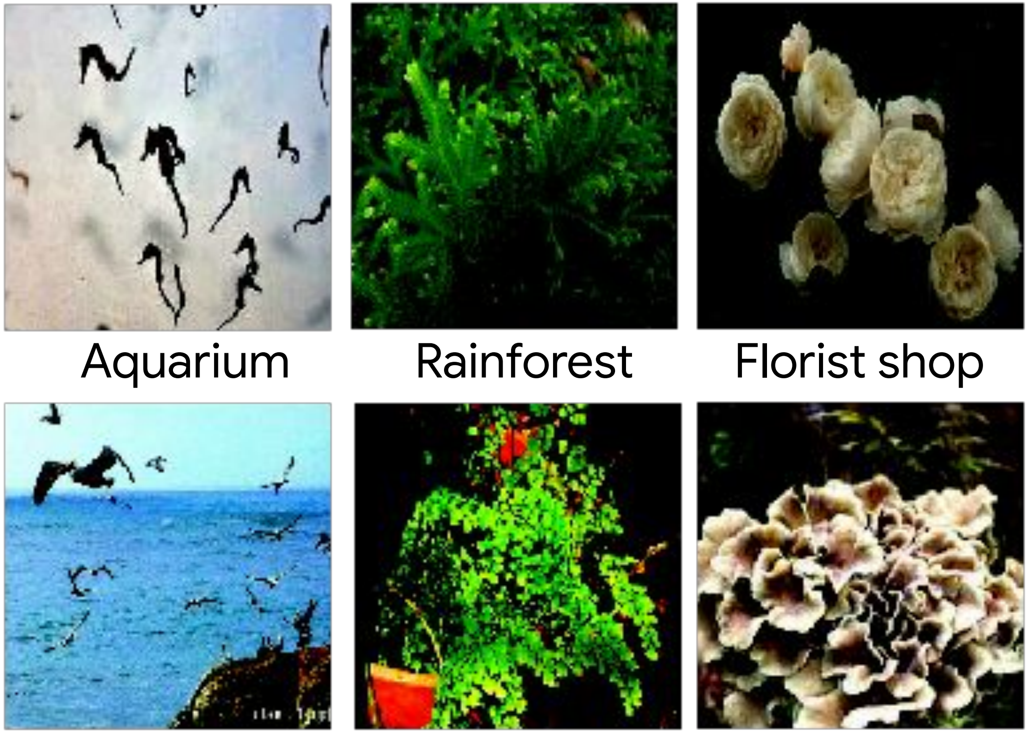
1 C

2

3

Highlights: Zero-Shot Open Set Recognition

[goo.gle/plex-paper]



OOD Input: Aquarium, Rainforest, Florist shop

Training Example: Pelican, Flowerpot, Hen-of-the-wood

Highlights: Label Uncertainty

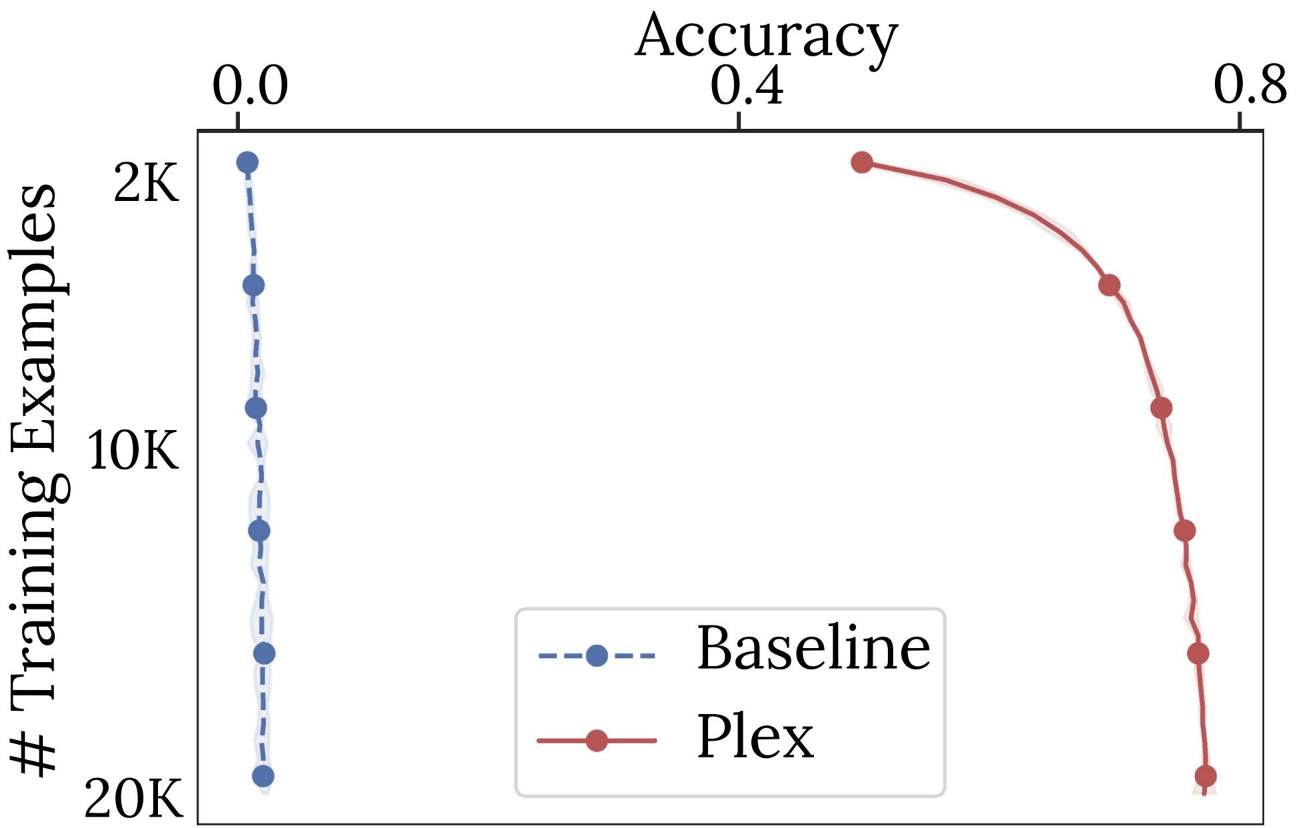
[goo.gle/plex-paper]

Label Uncertainty

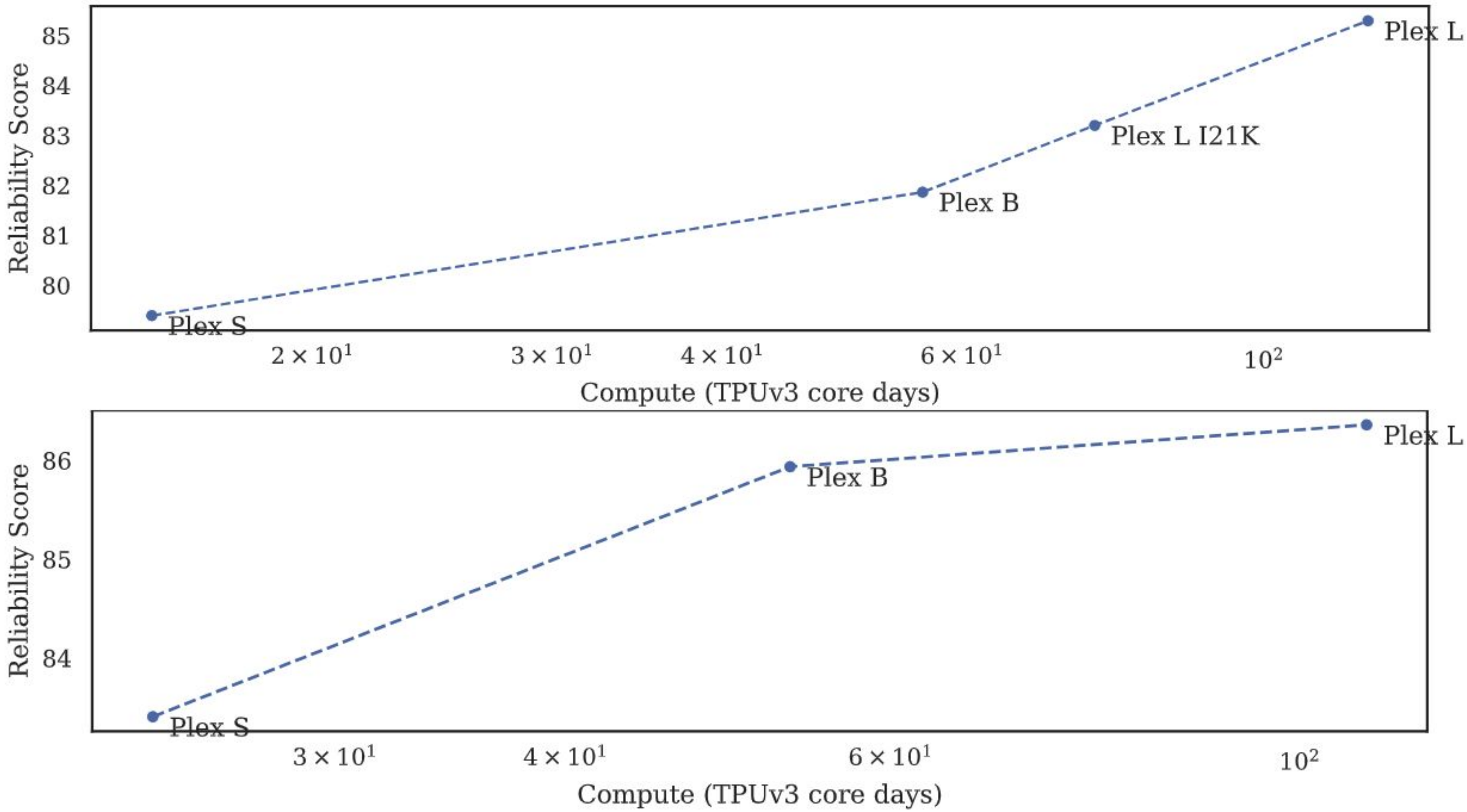
Truth $\begin{cases} 0.66 \text{ Strawberry} \\ 0.33 \text{ Fig} \end{cases}$

Highlights: Active Learning

[goo.gle/plex-paper]



Model Ablations: Impact of Scale (Vision (top) & Language (bottom))



* **Reliability score** is a normalized average over all task metrics: 139 for vision and 54 for language (see Appendix B of the paper).

Model Ablations: Impact of Uncertainty Model

[goo.gle/plex-paper]

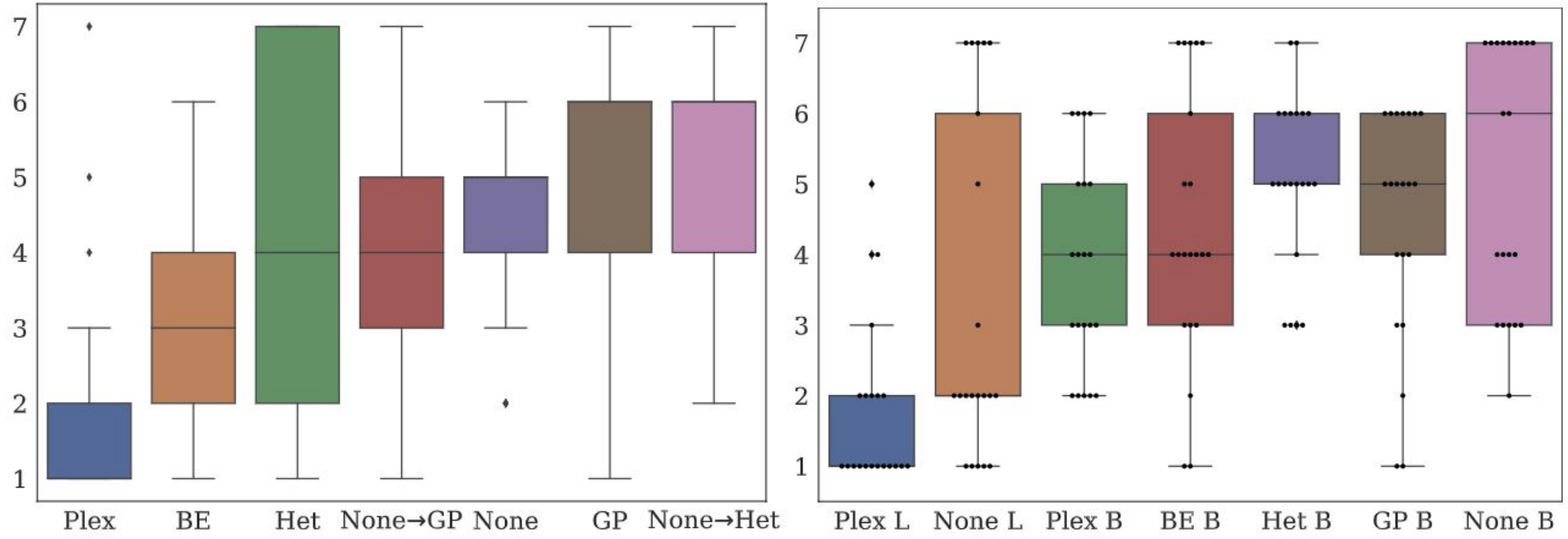
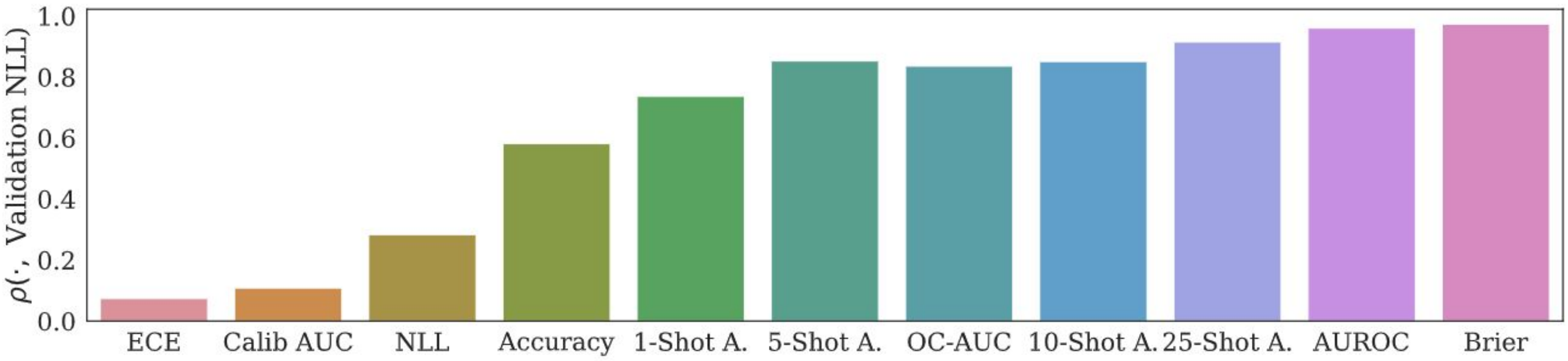


Figure 7: **(left)** Ranking of method ablations over 139 metrics on vision tasks and **(right)** 54 metrics on language tasks. Each model has a box plot of rankings (lower is better). Plex's combination of efficient ensembling and last-layer changes ranks best on average.

Relationship Between Reliability Tasks

[goo.gle/plex-paper]



Most tasks correlate highly ⇒
How well you fit pretraining data is a large predictor of downstream performance.

Wrapping up

Uncertainty Baselines

github.com/google/uncertainty-baselines

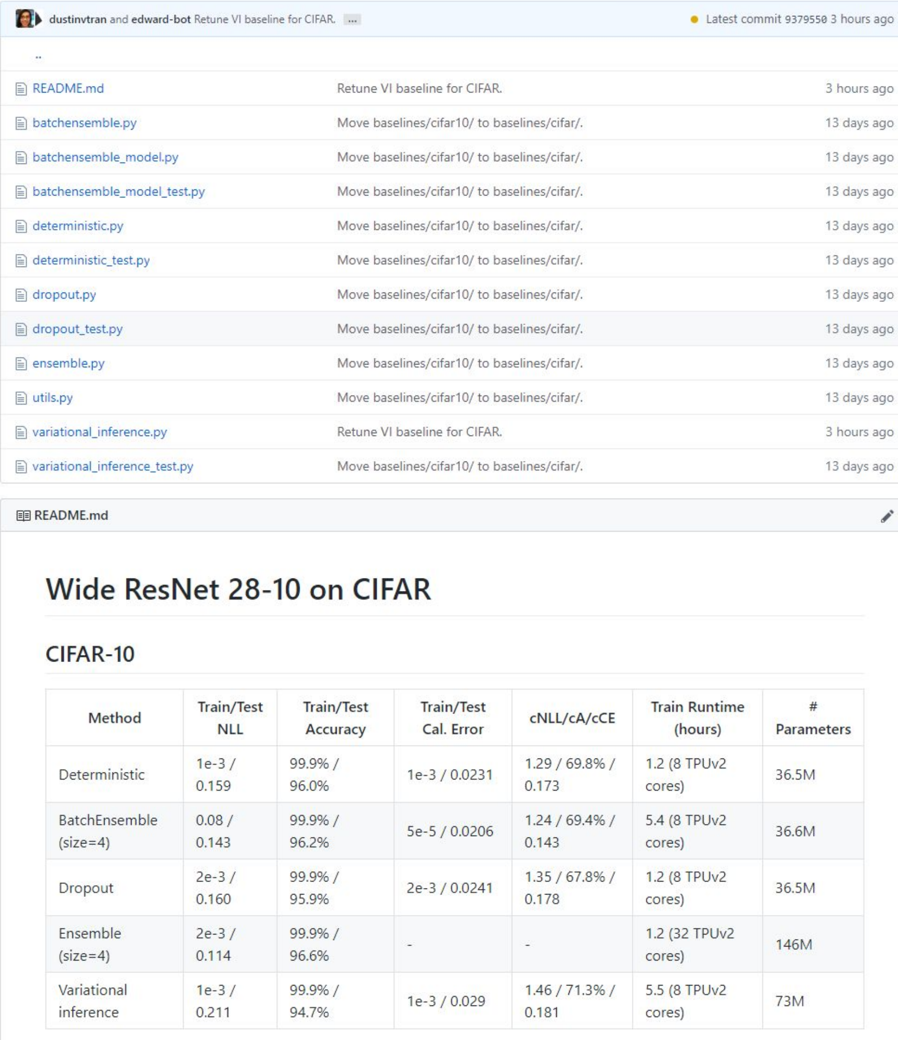
High-quality implementations of baselines in TensorFlow and JAX on a variety of tasks.

Ready for use: 65+ baselines across 9+ datasets and 14+ methods, including:

- Vision Transformer
- T5X-family
- Wide ResNet 28-10 on CIFAR
- ResNet-50 and EfficientNet on ImageNet
- BERT on Intent Detection & Toxicity Detection

Used across **15+** projects at Google.

Collaboration with OATML @ Oxford, unifying github.com/oatml/bdl-benchmarks.



dustinvtran and edward-bot Retune VI baseline for CIFAR. ... Latest commit 9379550 3 hours ago

..

README.md	Retune VI baseline for CIFAR.	3 hours ago
batchensemble.py	Move baselines/cifar10/ to baselines/cifar/.	13 days ago
batchensemble_model.py	Move baselines/cifar10/ to baselines/cifar/.	13 days ago
batchensemble_model_test.py	Move baselines/cifar10/ to baselines/cifar/.	13 days ago
deterministic.py	Move baselines/cifar10/ to baselines/cifar/.	13 days ago
deterministic_test.py	Move baselines/cifar10/ to baselines/cifar/.	13 days ago
dropout.py	Move baselines/cifar10/ to baselines/cifar/.	13 days ago
dropout_test.py	Move baselines/cifar10/ to baselines/cifar/.	13 days ago
ensemble.py	Move baselines/cifar10/ to baselines/cifar/.	13 days ago
utils.py	Move baselines/cifar10/ to baselines/cifar/.	13 days ago
variational_inference.py	Retune VI baseline for CIFAR.	3 hours ago
variational_inference_test.py	Move baselines/cifar10/ to baselines/cifar/.	13 days ago

README.md

Wide ResNet 28-10 on CIFAR

CIFAR-10

Method	Train/Test NLL	Train/Test Accuracy	Train/Test Cal. Error	cNLL/cA/cCE	Train Runtime (hours)	# Parameters
Deterministic	1e-3 / 0.159	99.9% / 96.0%	1e-3 / 0.0231	1.29 / 69.8% / 0.173	1.2 (8 TPUv2 cores)	36.5M
BatchEnsemble (size=4)	0.08 / 0.143	99.9% / 96.2%	5e-5 / 0.0206	1.24 / 69.4% / 0.143	5.4 (8 TPUv2 cores)	36.6M
Dropout	2e-3 / 0.160	99.9% / 95.9%	2e-3 / 0.0241	1.35 / 67.8% / 0.178	1.2 (8 TPUv2 cores)	36.5M
Ensemble (size=4)	2e-3 / 0.114	99.9% / 96.6%	-	-	1.2 (32 TPUv2 cores)	146M
Variational inference	1e-3 / 0.211	99.9% / 94.7%	1e-3 / 0.029	1.46 / 71.3% / 0.181	5.5 (8 TPUv2 cores)	73M

Takeaways

Google AI
Brain Team

- Uncertainty & robustness are critical problems in AI and machine learning.
- Best performance achieved by composing orthogonal techniques
    - Single model uncertainty
    - Ensembling multiple neural networks
    - Imposing inductive biases on representations
- Open questions
    - Understand relationship between different types of OOD shifts
    - What other tools do we need in the “composable” toolkit?
- Links to papers available in my webpage: http://www.gatsby.ucl.ac.uk/~balaji/

Thank you! Questions?

PLEX

Appendix

References

This list is intended just as a starting point for exploring other related work using Google Scholar or Connected papers. Feel free to email me if you think there's a reference that should be included here.

Google AI Brain Team

Survey papers

- **A Survey of Uncertainty in Deep Neural Networks**. J. Gawlikowski et al., *arXiv 2107.03342*.
- **A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges**. M. Abdar et al. *arXiv 2011.06225*

Bayesian neural networks

- **A practical Bayesian framework for backpropagation networks** D. MacKay Neural Computation 1992
- **Keeping Neural Networks Simple by Minimizing the Description Length of the Weights**. G. Hinton, D. Van Camp. COLT 1993.
- **An Introduction to Variational Methods for Graphical Models**. M. Jordan+. Machine Learning 1999.
- **Bayesian Learning for Neural Networks.** R. Neal. Technical Report 1994.
- **Bayesian Learning via Stochastic Gradient Langevin Dynamics**. M. Welling, Y. Teh. ICML 2011.
- **Weight Uncertainty in Neural Networks**. C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra. ICML 2015.
- **Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning** Y. Gal, Z. Ghahramani ICML 2016
- **Automatic Differentiation Variational Inference**. A. Kucukelbir, *D. Tran*, R. Ranganath, A. Gelman, D. M. Blei. JMLR 2017.
- **A Scalable Laplace Approximation for Neural Networks** H. Ritter, A. Botev, D. Barber ICLR 2018
- **Noise Contrastive Priors for Functional Uncertainty**. D. Hafner, *D. Tran*, T. Lillicrap, A. Irpan, J. Davidson. UAI 2019.
- **A Simple Baseline for Bayesian Uncertainty in Deep Learning** W. Maddox, T. Garipov, P. Izmailov, D. Vetrov, A. G. Wilson. NeurIPS 2019.
- **Practical Deep Learning with Bayesian Principles** K. Osawa, S. Swaroop, A.Jain, R. Eschenhagen, R. E. Turner, R. Yokota, M. E. Khan. NeurIPS 2019.
- **Efficient and Scalable Bayesian Neural Nets with Rank-1 Factors**. M. W. Dusenberry, G. Jerfel, Y. Wen, Y. Ma, J. Snoek, K. Heller, B. Lakshminarayanan, *D. Tran*. ICML 2020.

References

Google AI
Brain Team

Ensembles

- **Simple and scalable predictive uncertainty estimation using deep ensembles** *B. Lakshminarayanan*, A. Pritzel, C. Blundell. NeurIPS 2017.
- **BatchEnsemble: An Alternative Approach to Efficient Ensemble and Lifelong Learning**. Y. Wen, *D. Tran*, J. Ba. ICLR 2020.
- **Why Aren't Bootstrapped Neural Networks Better?** J. Nixon, D. Tran and B. Lakshminarayanan ICBINB workshop @ NeurIPS 2020.
- **Training independent subnetworks for robust prediction**. M. Havasi, R. Jenatton, S. Fort, J. Z. Liu, J. Snoek, B. Lakshminarayanan, A. M. Dai, D. Tran ICLR 2021.

Understanding marginalization

- **Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data**. G. K. Dziugaite, D. M. Roy. UAI 2017.
- **Deep ensembles: A loss landscape perspective**. S. Fort, H. Hu, *B. Lakshminarayanan*. *arXiv 1912.02757*.
- **Bayesian Deep Learning and a Probabilistic Perspective of Generalization** A. G. Wilson and P. Izmailov arXiv 2002.08791

Gaussian processes and Neural Tangent Kernel

- **Deep Neural Networks as Gaussian Processes.** J. Lee, Y. Bahri, R. Novak, S. Schoenholz, J. Pennington, J. Sohl-Dickstein, ICLR 2018.
- **Neural Tangent Kernel: Convergence and Generalization in Neural Networks.** A. Jacot, F. Gabriel, C. Hongler. NeurIPS 2018.
- **Approximate Inference Turns Deep Networks into Gaussian Processes** M. Emtiyaz Khan, Alexander Immer, Ehsan Abedi, M. Korzepa NeurIPS 2019
- **Bayesian Deep Ensembles via the Neural Tangent Kernel** B. He, *B. Lakshminarayanan* and Y.W. Teh *NeurIPS 2020*
- **Exploring the Uncertainty Properties of Neural Networks' Implicit Priors in the Infinite-Width Limit**. B. Adlam, J. Lee, L. Xiao, J. Pennington and J. Snoek link

References

Practical guidance

- See the codebases! E.g. https://github.com/google/uncertainty-baselines
- **Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift**. Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D Sculley, S. Nowozin, J. Dillon, *B. Lakshminarayanan*, *J. Snoek*. NeurIPS 2019.
- **Simple, Distributed, & Accelerated Probabilistic Programming**. *D. Tran*, M. Hoffman, D. Moore, C. Suter, S. Vasudevan, A. Radul, M. Johnson, R. A. Saurous. NeurIPS 2018.
- **Bayesian Layers: A Module for Neural Network Uncertainty**. *D. Tran*, M. W. Dusenberry, M. van der Wilk, D. Hafner. NeurIPS 2019.

Data Augmentation and Invariances

- **Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty** *D. Hendrycks, M. Mazeika, S. Kadavath, D. Song NeurIPS 2019*
- **AugMix: A simple data processing method to improve robustness and uncertainty**. D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, *B. Lakshminarayanan*. *ICLR 2020*
- **Combining Ensembles and Data Augmentation can Harm your Calibration**. *Y. Wen, G. Jerfel, R. Muller, M. Dusenberry, J. Snoek, B. Lakshminarayanan and D. Tran. ICLR 2021*

Calibration

- **On calibration of modern neural networks** C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger *ICML 2017*
- **Revisiting the Calibration of Modern Neural Networks**. M. Minderer, J. Djolonga, R. Romijnders, F. Hubis, X. Zhai, N. Houlsby, D. Tran, M. Lucic *arXiv 2106.07998*.
- **Measuring Calibration in Deep Learning**. J. Nixon, M. Dusenberry, L. Zhang, G. Jerfel, *D. Tran*. arXiv 1904.01685

Google AI Brain Team

References

Google AI Brain Team

Proper Scoring Rules, Types of Uncertainty

- **Strictly Proper Scoring Rules, Prediction and Estimation**, *Gneiting & Raftery, JASA 2007*
- **What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?** *A. Kendall, Y. Gal NeurIPS 2017*
- **Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods** *E. Hüllermeier, W. Waegeman* arXiv 1910.09457

Deep Generative Models and Hybrid models

- **Hybrid models with deep and invertible features** E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, *B. Lakshminarayanan*. *ICML 2019*.
- **Do deep generative models know what they don't know?** E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, *B. Lakshminarayanan*. *ICLR 2019*.
- **Likelihood ratios for out-of-distribution detection**. J. Ren, P. Liu, E. Fertig, J. Snoek, R. Poplin, M. DePristo, J. Dillon, *B. Lakshminarayanan*. *NeurIPS 2019*.
- **Detecting out-of-distribution inputs to deep generative models using a test for typicality.** E. Nalisnick, A. Matsukawa, Y. W. Teh, *B. Lakshminarayanan*. *arXiv 1906.02994*.
- **Density of States Estimation for Out-of-Distribution Detection** W. R. Morningstar, C. Ham, A. G. Gallagher, B. Lakshminarayanan, A. A. Alemi, J. V. Dillon AISTATS 2021

Detecting Out-of-Distribution Inputs

- **A Baseline for Detecting Misclassified & Out-of-Distribution Examples in Neural Networks** D. Hendrycks, K. Gimpel ICLR 2017
- **A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks** K. Lee, K. Lee, H. Lee, J. Shin NeurIPS 2018
- **Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks** S. Liang, Y. Li, R. Srikant ICLR 2018
- **Deep Anomaly Detection with Outlier Exposure** D. Hendrycks, M. Mazeika, T. Dietterich ICLR 2019
- **Exploring the Limits of Out-of-Distribution Detection**. S. Fort, J. Ren, B. Lakshminarayanan *NeurIPS 2021*.