# Uncertainty in Deep Learning
# &
# The case of Bayesian Deep Learning


Presenter: Son Nguyen

*Machine Learning Group, VinAI Research*

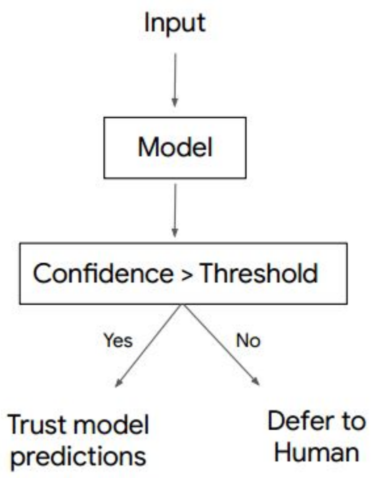
# Why need uncertainty in Deep Learning

❖ **Uncertainty estimation:** critical problem (**applicable <-- reliable**) in Intelligent Systems

- provide **confidence** along with prediction: *the model knows what it doesn't know*
- go beyond **accuracy** regime: toward **model calibration** in Deep Learning (DL)

❖ **Applications**:

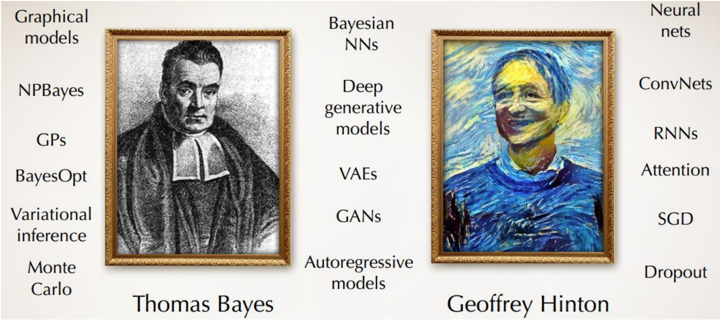
- Safety, Trustworthy systems: *autonomous driving, medical diagnosis, and meteorological forecasting.*
- Active learning, Continual learning, Reinforcement learning, Bayesian optimization, Decision making: *trade off exploration-exploitation, stability-plasticity, memorization-adaptation*



# What Bayesian Deep Learning

❖ **Bayesian Deep Learning (BDL)**: general principle, structural probabilistic approach

- **intersection** of Bayesian method and deep learning
- *Bayesian neural nets, deep latent variable models, and related learning techniques* are particular treatments of BDL.
- Advances in BDL: [Bayesian Deep Learning workshops]
    - In supervised tasks, BDL provide considerable improvements in *accuracy and calibration* compared to standard training, while retaining scalability.




❖ **A main goal**: exploring a renowned class of BDL - Bayesian neural nets (BNNs)

- the core direction promoting the research of uncertainty quantification in DL
- but, has many controversies in the community

# Content

## A. Uncertainty in Deep Learning

1. Background
2. Main approaches
3. The state-of-the-art and a unified perspective
4. Some potential research

## B. Bayesian neural network and its controversies

1. Why Bayesian neural nets
2. Expressive or simple approximate posterior distribution
3. Tempered or original true posterior distribution
4. Informative or vague prior distribution

*This presentation involves various works of Yarin Gal (OATML), Andrew G. Wilson (NYU), B. Lakshminarayanan (Google), Dustin Tran (Google), Max Welling (UvA) and many others.*

# Content

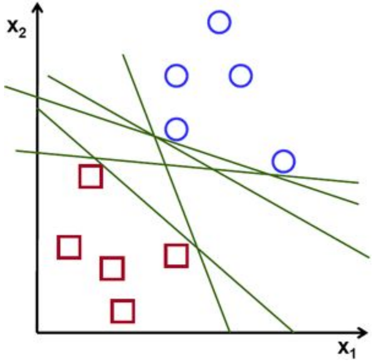
## A. Uncertainty in Deep Learning

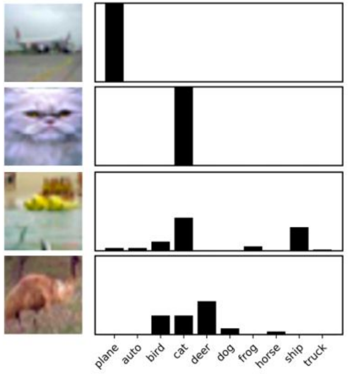
1. Background
   a. Sources of uncertainty
   b. Disentangle types of uncertainty
   c. Metrics for uncertainty quantification
   d. Common criticism of traditional neural nets uncertainty
2. Main approaches
3. The state-of-the-art and a unified perspective
4. Some potential research

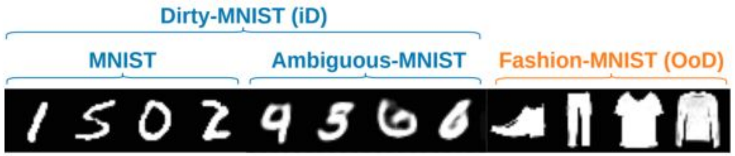
# Background

❖ **Source of uncertainty: [NeurIPS-17]**

| | |
|---|---|
| Model uncertainty, a.k.a **epistemic uncertainty** | capture our ignorance about which model generated our collected data |
| | incurred by *lack of training data, imbalanced/sparse data, out-of-distribution data* |
| | **reducible** with more data (vanish in the limit of infinite data) |
| Data uncertainty, a.k.a **aleatoric uncertainty** | capture noise inherent in the data |
| | caused by *inherent noise, ambiguous/missing data, human bias* |
| | **irreducible** with more data |



# Background

## ❖ Disentangle types of uncertainty

- Disentangling and reasoning about uncertainty is critical, but **non-trivial**, for applications:
  - *active learning [NeurIPS-19]*
  - *out-of-distribution detection*
  - *semantic segmentation [NeurIPS-17]*
  - *fraud detection, forecast*

# Background

❖ **Disentangle types of uncertainty [UAI-18]**

- **epistemic** and **aleatoric** uncertainty are *distinguishable under Bayesian models:*

Model parameters $W$ governed by a prior $p(W)$, and $p(W|\mathcal{D})$ is a posterior given the training data $\mathcal{D}$

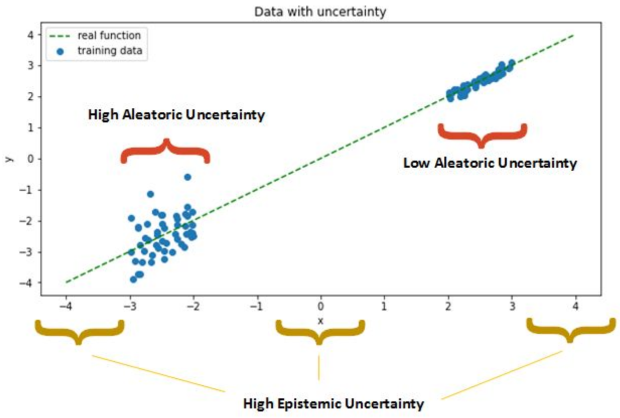
The **predictive distribution** for a new datapoint $(x, y)$ is: $p(y|x,\mathcal{D}) = \mathbb{E}_{p(W|\mathcal{D})} p(y|x,W)$

  - the predictive entropy $\mathbb{H}[y|x,\mathcal{D}]$ of $p(y|x,\mathcal{D})$ is defined by **predictive uncertainty**
  - predictive uncertainty is **total uncertainty** of epistemic and aleatoric uncertainty.

$$\underbrace{\mathbb{H}[Y|x,\mathcal{D}]}_{\text{predictive}} = \underbrace{\mathbb{I}[Y;w|x,\mathcal{D}]}_{\text{epistemic}} + \underbrace{\mathbb{E}_{p(W|\mathcal{D})}[\mathbb{H}[Y|x,w]]}_{\text{aleatoric (for iD } x)}$$

In Bayesian linear regression case:

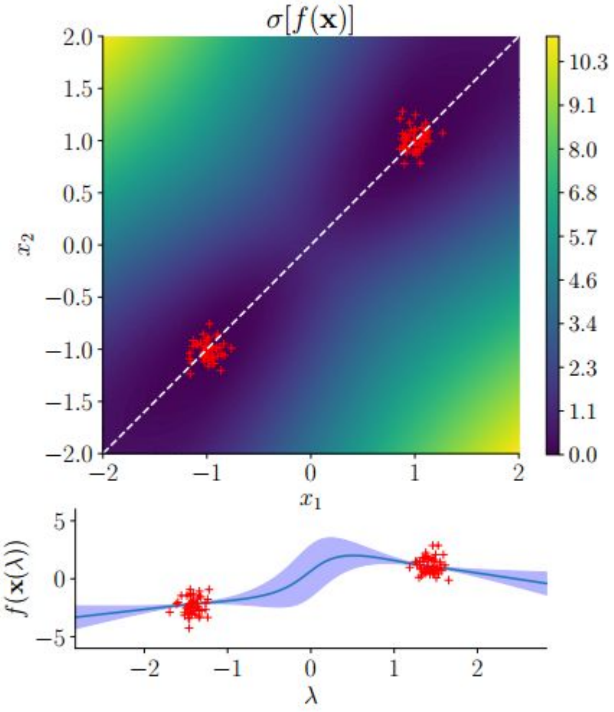
$$\mathbb{V}(y|x,\mathcal{D}) = \phi(x)^T \Sigma \phi(x) + \sigma^2$$

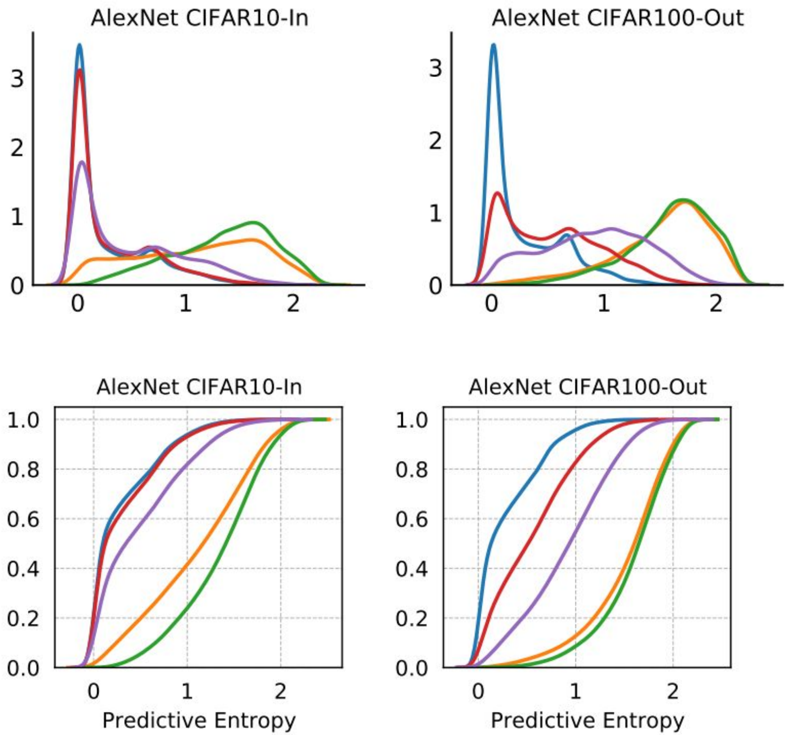


# Background

❖ **Metrics for uncertainty quantification**

- How to represent uncertainty: **heat map, predictive variance, predictive entropy** (PDF, CDF).



# Background

❖ **Metrics for uncertainty quantification**

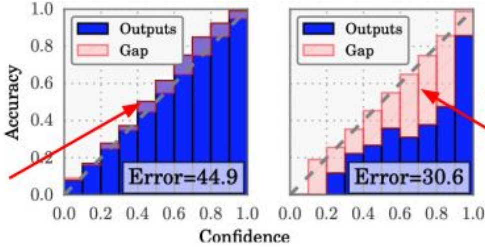
- How to measure the quality of uncertainty:
  - **predictive log-likelihood:** $\mathbb{E}_{x\sim\mathcal{D}} \log[\mathbb{E}_{p(W|\mathcal{D})} p(y|x, W)]$
  - **calibration error (CE) [ICML-17]:** suppose a model predict a class y with probability $\hat{p}$

$$\mathrm{CE} = \left|\mathrm{Prob}(Y = y|\hat{p} = p) - p\right|$$

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

Confidence < Accuracy

=> Underconfident




Confidence > Accuracy

=> Overconfident

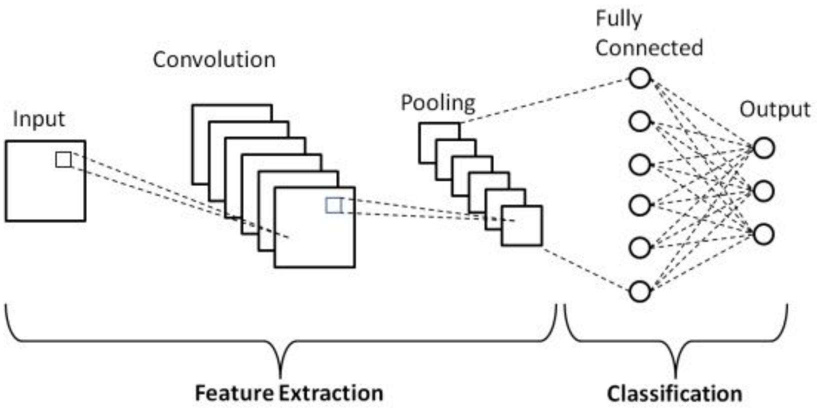
$$\mathrm{SCE} = \frac{1}{K} \sum_{k=1}^{K} \sum_{b=1}^{B} \frac{n_{bk}}{N} |\mathrm{acc}(b, k) - \mathrm{conf}(b, k)|$$

with $\mathrm{acc}(b, k)$ and $\mathrm{conf}(b, k)$ are the accuracy and confidence of bin b for class label k

# Background

❖ **Common criticism of traditional neural nets uncertainty**

- **Trend**: larger and more accurate models produce poorly calibrated predictions.
- **Disentangle** epistemic and aleatoric uncertainty is non-trivial: use **softmax entropy** in general.
- **Softmax** deterministic neural nets can not capture epistemic uncertainty: **feature collapse** *(theory and empirical results)* --> extractor can map OOD sample to iD regions in feature space (*local constant representation*). [ICML-20]




When training using empirical risk minimisation, *features not relevant to classification accuracy* can simply be ignored by the feature extractors.

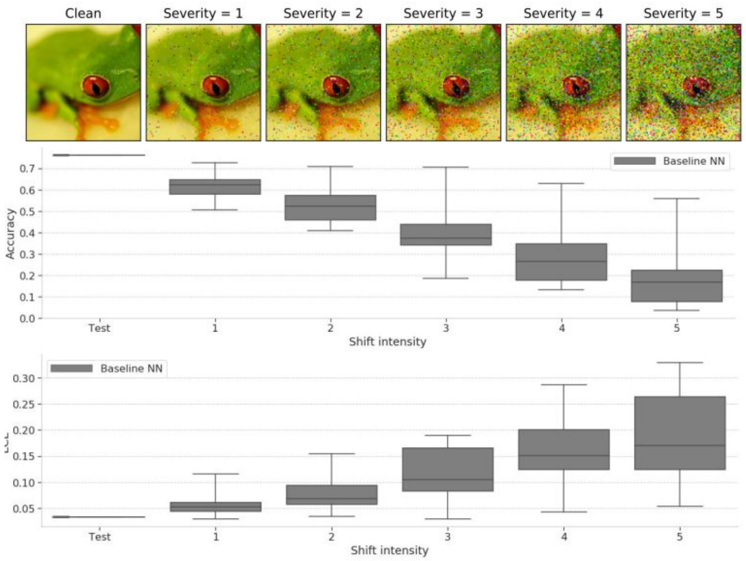
# Background

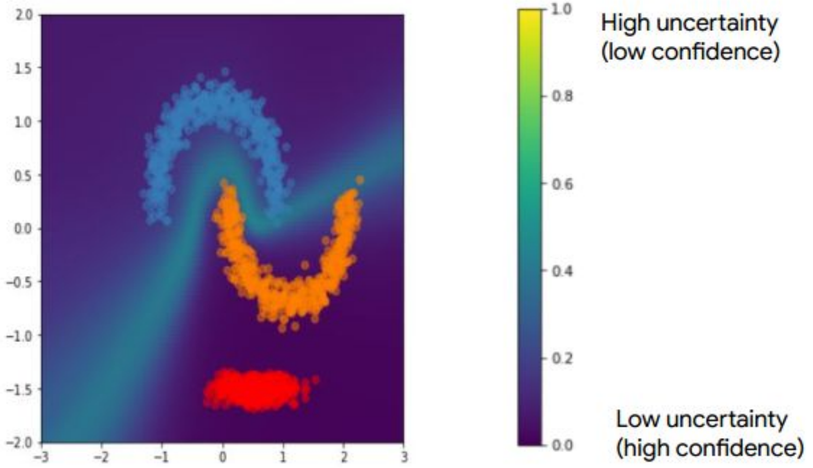
❖ **Common criticism of traditional neural nets uncertainty**

- NNs do not generalize well under **distribution shift.**

  but, NNs do not know when they do not know.

- Models assign high confidence predictions to OOD data



# Summary

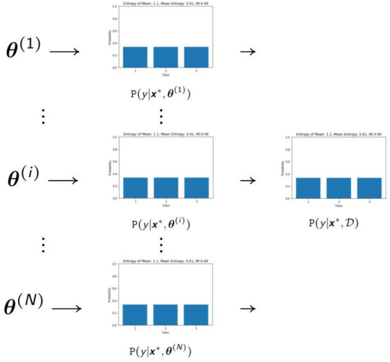
- **Source of uncertainty:** epistemic (lack of data) and aleatoric (noise inherent)
- **Disentangle** epistemic and aleatoric is non-trivial, but possible with Bayesian models:

$$\underbrace{\mathbb{H}[Y|x,\mathcal{D}]}_{\text{predictive}} = \underbrace{\mathbb{I}[Y;w|x,\mathcal{D}]}_{\text{epistemic}} + \underbrace{\mathbb{E}_{p(W|\mathcal{D})}[\mathbb{H}[Y|x,w]]}_{\text{aleatoric (for iD } x)}$$

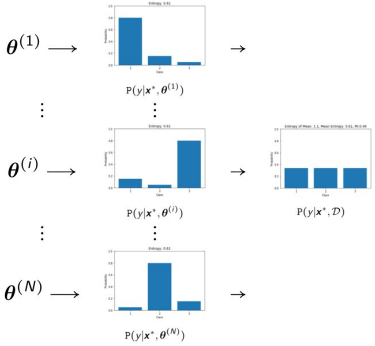
In *Bayesian linear regression* case:

$$\mathbb{V}(y|x,\mathcal{D}) = \phi(x)^T \Sigma \phi(x) + \sigma^2$$

A connection with *bias-variance trade-off:* $y = f(x) + \epsilon$




high aleatoric




high epistemic

$$\mathbb{E}_y \mathbb{E}_{p(W|\mathcal{D})} \left[ y - \widehat{f}(x,W) | \mathcal{D} = \mathcal{D}_{train} \right]^2 = \left( f(x) - \mathbb{E}_{p(W|\mathcal{D})} \left[ \widehat{f}(x,W) \right] \right)^2 + \mathbb{V}_{p(W|\mathcal{D})} \left[ \widehat{f}(x,W) \right] + \sigma^2$$

- **Measure** deep network models: **predictive accuracy** (generalization), **likelihood/ECE/SCE** (model calibration)
- **Criticisms** of traditional deep learning uncertainty: *poor generalization under distribution shift, uncalibrated and overconfident prediction, inability to capture epistemic uncertainty* ***(feature collapse)***

# Content

## A. Uncertainty in Deep Learning

1. Background
2. Main approaches
    a. Bayesian neural nets
    b. Ensemble methods
    c. Deterministic uncertainty estimation
3. The state-of-the-art and a unified perspective
4. Some potential research

# Main approaches

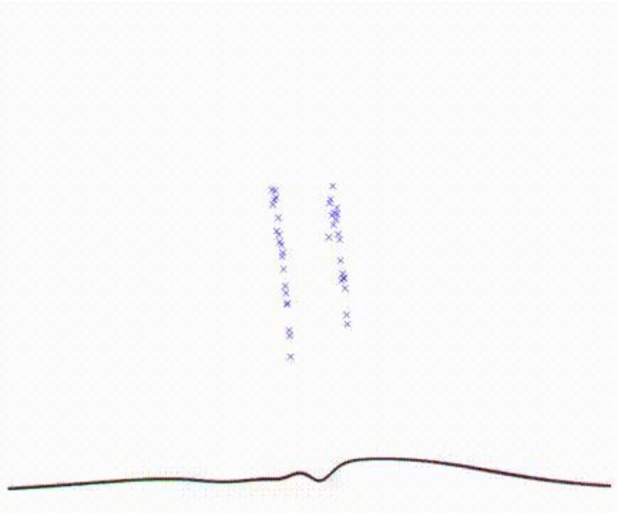
## ❖ Bayesian neural nets

- treat weight parameters $W$ as a random variable and impose a prior distribution $p(W)$
- infer a posterior distribution over $W$ instead of point estimation:

$$p(\mathbf{W}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{W})p(W)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\mathbf{W})p(W)}{\int p(\mathcal{D}|\mathbf{W})p(\mathbf{W})}$$

- At test time: predictive distribution is approximated via **MC sampling**:

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) = \int p(\mathbf{y}|\mathbf{x}, \mathbf{W})p(\mathbf{W}|\mathcal{D})d\mathbf{W} = \frac{1}{S}\sum_{s=1}^{S} p(\mathbf{y}|\mathbf{x}, \mathbf{W}^{(s)})$$



# Main approaches

❖ **Bayesian neural nets**

BNN posterior $p(W|\mathcal{D})$ : *intractable, very high dimensional, complicated structure* --> **approximate inference**

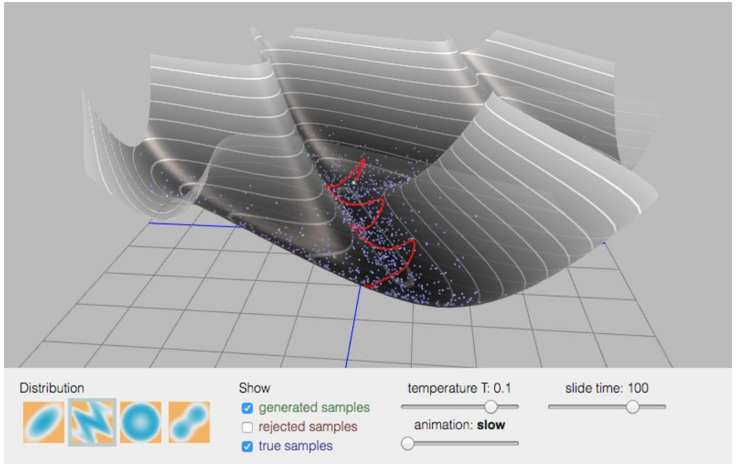
- **Gradient-based stochastic approximation**:
  - energy-based perspective
  - simulate **dynamical systems** whose **stationary distribution** as desired target distribution
  - the true posterior samples is generated via discretizing differential equations describing those dynamics

**Methods:**

- Hamiltonian Monte Carlo (HMC): gold standard
- Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) (ICML-14)
- Stochastic Gradient Langevin Dynamics (SGLD) (ICML-12)

**Pros and Cons:** high fidelity approximation,

but large complexity, many potential biases



# Main approaches

❖ **Bayesian neural nets**

BNN posterior $p(W|\mathcal{D})$: *intractable, very high dimensional, complicated structure* --> **approximate inference**

- **Deterministic approximation:** local approximation
  - **Laplace approximation:** $p(W|\mathcal{D}) = \mathcal{N}(W_{MAP}, H^{-1})$ with $H = \partial^2 \log p(y|x, W)/\partial W^2 + \lambda I$
  - **Variational inference:** employ a **parametric** variational distribution $q_\phi(\mathbf{W})$ and minimize $\mathbb{D}_{KL}(q_\phi(\mathbf{W})\|p(\mathbf{W}|\mathcal{D}))$

    equivalent to maximizing variational lower bound:

    $$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\mathbf{W})} \log p(\mathcal{D}|\mathbf{W}) - \mathbb{D}_{KL}(q_\phi(\mathbf{W})\|p(\mathbf{W}))$$

    - **Mean-field VI:** $q_\phi(\mathbf{W})$ is factorized distribution (e.g diagonal Gaussian)
    - **Dropout inference:** MC Dropout, Variational Gaussian Dropout --> **complementary** benefits
    - **Subspace inference** (UAI-19): inspired by ***effective dimensionality / intrinsic dimension*** in deep learning

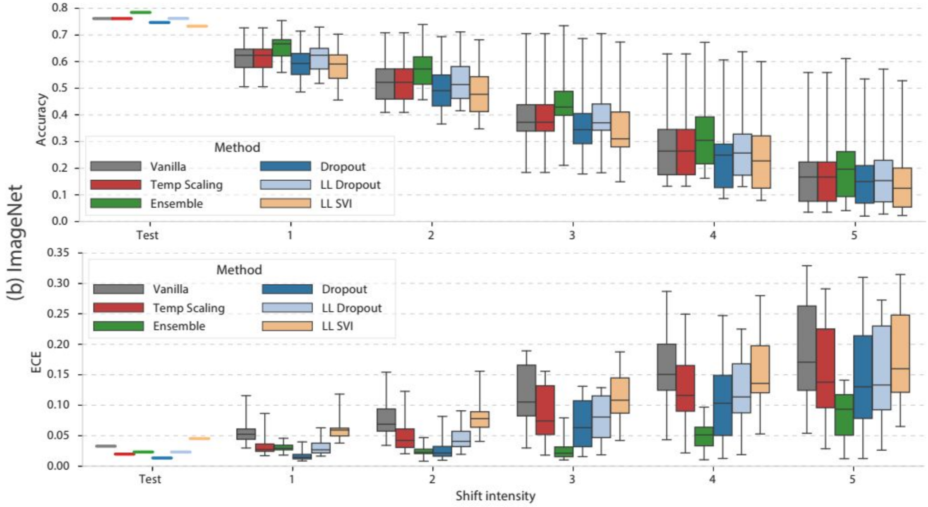
    $$\mathcal{S} = \{W|W = \widehat{W} + z_1 v_1 + \ldots + z_K v_K\} = \{W|W = \widehat{W} + Pz\}$$

    ****sub-network (ICML-21):** $p(\mathbf{W}|\boldsymbol{y}, \boldsymbol{X}) \approx p(\mathbf{W}_S|\boldsymbol{y}, \boldsymbol{X}) \prod_r \delta(\mathrm{w}_r - \mathrm{w}_r^*) \approx q(\mathbf{W}_S) \prod_r \delta(\mathrm{w}_r - \mathrm{w}_r^*)$

# Main approaches

❖ **Ensemble methods**

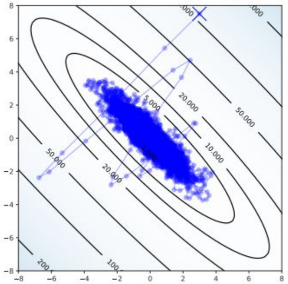
- **Deep ensemble (NeurIPS-16)**: training (regularized) MLE with different random seeds and averaging final score
  - inspired by classical ensemble methods: bootstrap, bagging, boosting
  - loss landscape is highly non-convex --> different local optima --> explore the diversity from multimodality.
  - very simple, but work surprisingly well in practice



# Main approaches

❖ **Ensemble methods**

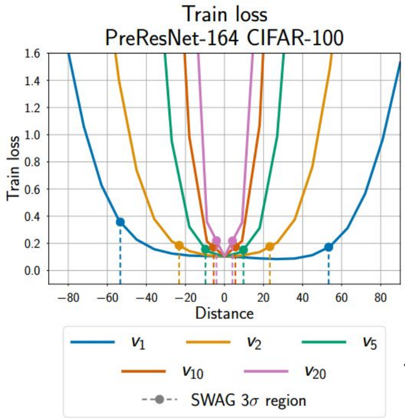
- **Stochastic Weight Averaging Gaussian (SWAG) (NeurIPS-19)**
  - **Motivated by the theory:** SGD with constant learning rate simulates a Markov chain with a stationary distribution --> SGD iterations is approximately sampling from a Gaussian distribution (JMLR-17)



  - Utilize SGD iterations $\{W_i\}_{i=1}^{T}$ to empirically estimate ***first-two moments*** of a Gaussian: $p(W|\mathcal{D}) = \mathcal{N}(\mu, \Sigma)$

$$\mu = \frac{1}{T}\sum_t W_t \qquad \Sigma = \frac{1}{T-1}\sum_t \left(W_t - \overline{W}_t\right)\left(W_t - \overline{W}_t\right)^T \quad \boxed{\left(+\frac{1}{T}\mathrm{diag}\left(\frac{1}{T}\sum_{i=1}^{T} W_i^2 - \mu^2\right)\right)}$$

  - **Properties:**
    - require: SGD with ***large constant*** or ***cyclical learning rates***
    - practical runtime ~ SGD training
    - Averaging Weights Leads to Wider Optima and Better Generalization (**SWA** *PyTorch lib*) (ICML-18)
    - captures the local geometry of the posterior surprisingly well



# Main approaches

❖ **Deterministic uncertainty estimation (DUE)**

**Motivation**: overcome limitations of softmax neural nets uncertainty

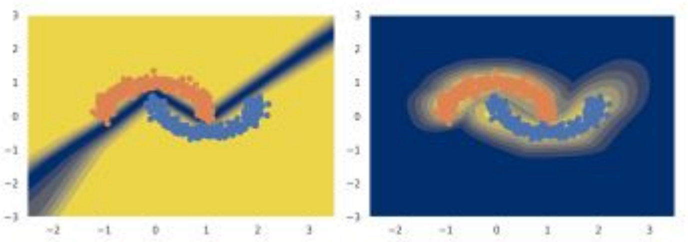
--> using only single forward-pass

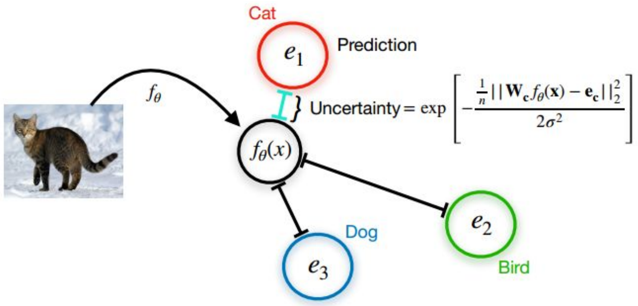
# Main approaches

- ❖ **Deterministic uncertainty estimation (DUE)**
  - **DUE with RBF network. (ICML-20)**
    - classes represented by **centroids**
    - predictive uncertainty computed via **RBF kernel**

      --> better than Deep ensemble uncertainty
    - use **exponential moving average** update to stabilize training

      --> achieve competitive accuracy softmax models.
    - alleviate **feature collapse** with two-side **Gradient penalty**
      - sensitivity: capture changes in inputs
      - smoothness: optimization & generalization
    - **What about softmax nets + enforcing-sensitivity ?**






$$K_c(f_\theta(\mathbf{x}), \mathbf{e}_c) = \exp\left[-\frac{\frac{1}{n}||\mathbf{W}_c f_\theta(\mathbf{x}) - \mathbf{e}_c||_2^2}{2\sigma^2}\right]$$

$$L(\mathbf{x}, \mathbf{y}) = -\sum_c y_c \log(K_c) + (1 - y_c)\log(1 - K_c)$$

$$\lambda \cdot \left[||\nabla_{\mathbf{x}} \textstyle\sum_c K_c||_2^2 - L\right]^2 \longrightarrow \; L_1||\mathbf{x}_1 - \mathbf{x}_2||_I \leq ||K_c(\mathbf{x}_1) - K_c(\mathbf{x}_2)||_F \leq L_2||\mathbf{x}_1 - \mathbf{x}_2||_I$$

# Main approaches

❖ **Deterministic uncertainty estimation (DUE)**

- **DUE with softmax nets + inductive bias + feature-space density (arXiv-21).**
  - *gradient penalty, spectral normalization* are appropriate inductive biases enforcing sensitivity
  - penalize **spectral normal** of deterministic networks weights, then:
    - **softmax entropy** can capture aleatoric uncertainty, but can not estimate epistemic uncertainty
    - use **feature-space density** $q(z)$, with $z = f_\theta(x)$ to capture epistemic uncertainty
    - combine feature-space density and the softmax entropy via Gaussian Discriminant Analysis (GDA)

$$q(y, z) = q(y)q(z|y)$$ ----> disentangle epistemic and aleatoric uncertainty


(a) Dirty-MNIST (iD) and Fashion-MNIST (OoD) (b) Softmax entropy (MNIST vs Ambiguous-MNIST)

(c) Softmax entropy (d) Feature space density

# Content

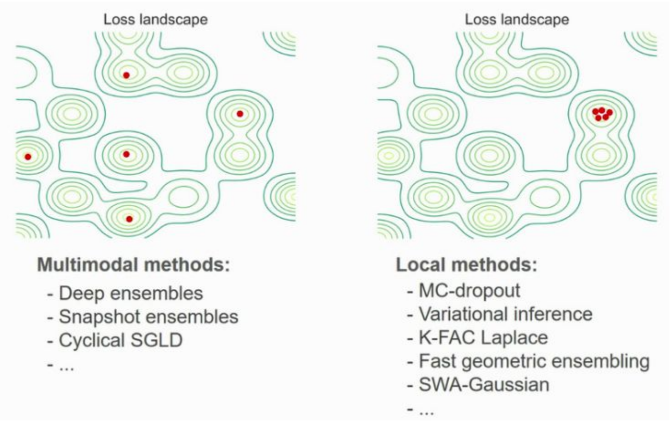
## A. Uncertainty in Deep Learning

1. Background
2. Main approaches
3. The state-of-the-art and a unified view
   a. Deep ensemble and variants
   b. Bayesian model averaging
4. Some potential research

# The state-of-the-art and a unified view

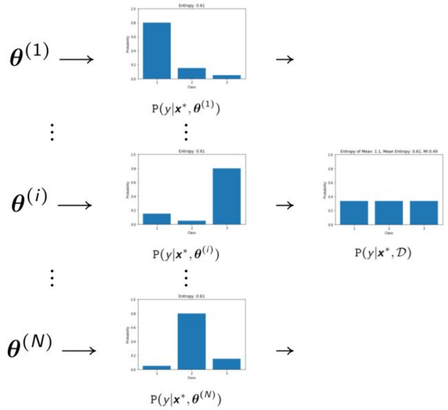
❖ **Deep ensemble and functional perspective (arXiv-20)**

- **Consistent experimental results**: Deep ensemble
  - very simple, but work surprisingly well in practice
  - outperforms SWAG, practical BNNs approximations (MFVI, MC Dropout), *particularly under dataset shift.*
  - but has much *computational overhead*




- **A functional perspective:**
  - **desiderata** from ensembling for a good approximation of **predictive distribution:**

    **high-performing** but **diverse**
    - similar predictions will be redundant in the model averaging
    - crucial for quantifying epistemic uncertainty [NeurIPS-17]
  - **Main point:** deep ensembles tend to explore **diverse modes in functional space.**

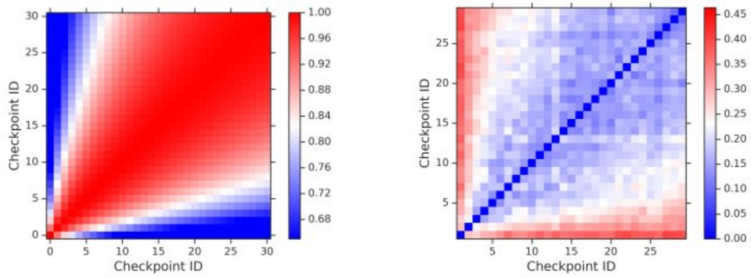


# The state-of-the-art and a unified view

❖ **Deep ensemble and functional perspective**

- Similarity of functions *within and across* randomly initialized trajectories

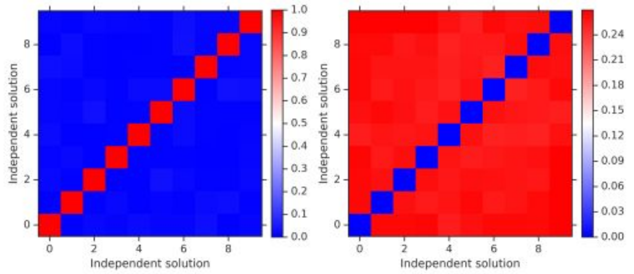
**SGD single trajectory**



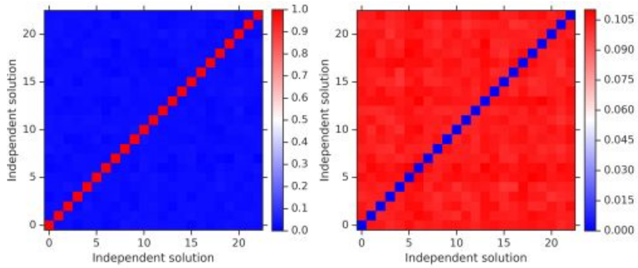

(a) Cosine similarity of weights (b) Disagreement of predictions

$$\cos(W_1, W_2) = \frac{W_1^T W_2}{\|W_1\| * \|W_2\|} \quad \frac{1}{N} \sum_{n=1}^{N} [f(x_n; W_1) \neq f(x_n; W_2)]$$

**Deep Ensemble**




(a) Results using *SmallCNN*

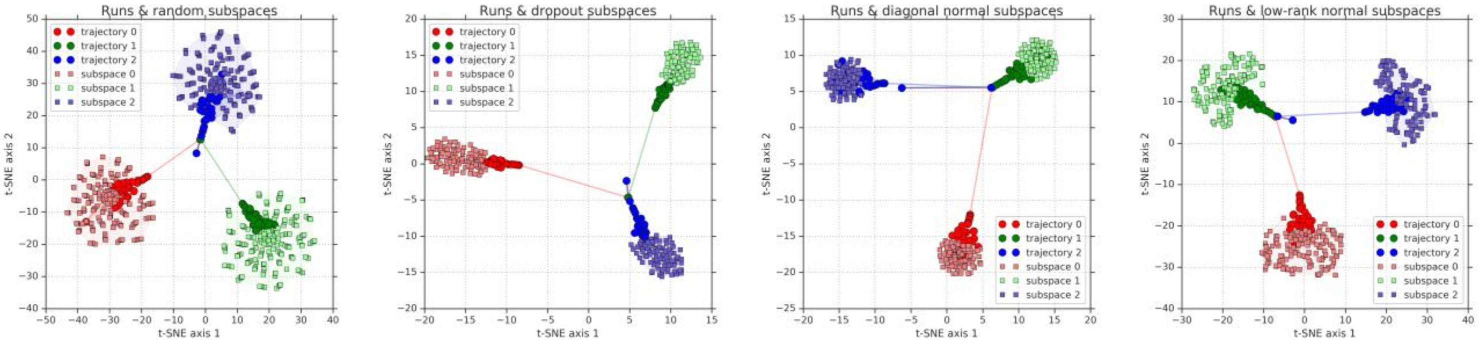



(b) Results using *ResNet20v1*

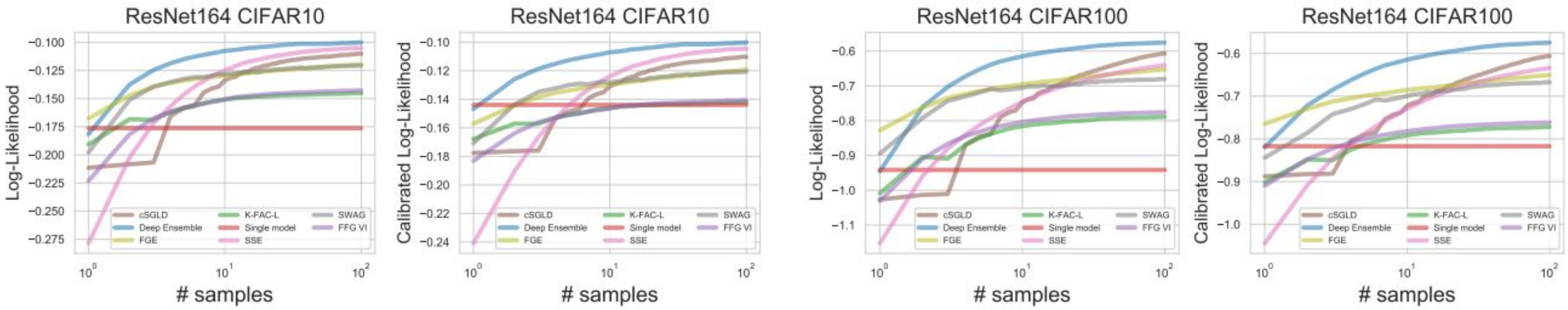
# The state-of-the-art and a unified view

## ❖ Deep ensemble and functional perspective

- Similarity of functions of **local approximations** from each trajectory and across trajectories



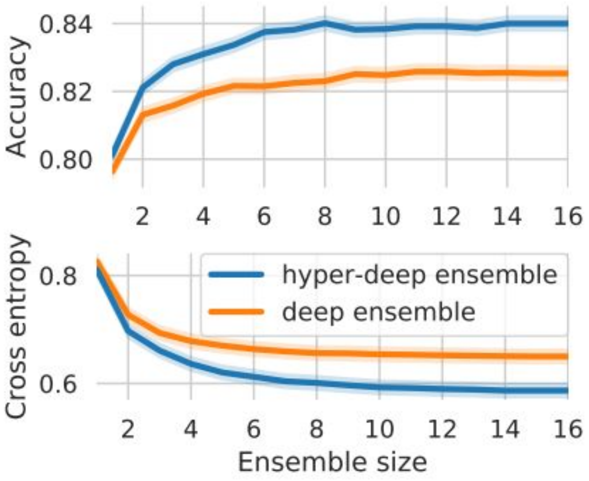

- Accuracy as a function of ensemble size



# The state-of-the-art and a unified view

❖ **Several variants of deep ensemble**

- **Hyperparameter ensembles (NeurIPS-20)**: random search over different hyperparameters

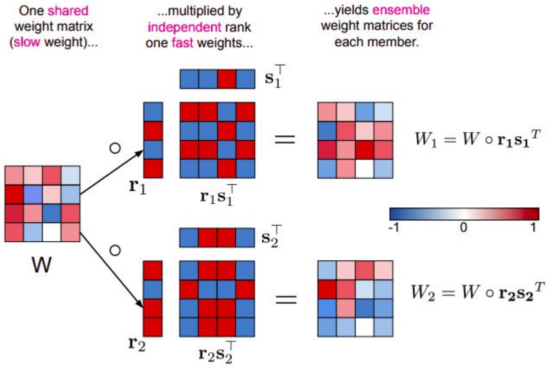


# The state-of-the-art and a unified view

❖ **Several variants of deep ensemble:** *inspired by <u>sharing parameters</u>*

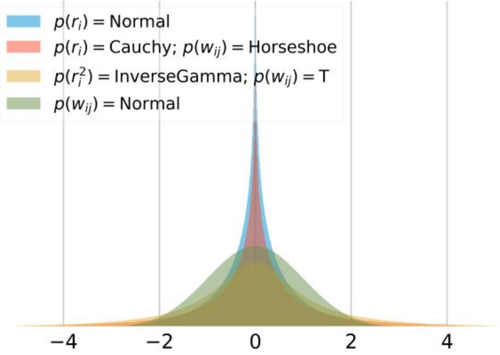
- **Batch ensemble (<u>ICLR-20</u>):** efficient ensembles by **sharing parameters**

$$
\begin{aligned}
y_n &= \phi\left(\overline{W}_i^\top x_n\right) \\
&= \phi\left(\left(W \circ r_i s_i^\top\right)^\top x_n\right) \\
&= \phi\left(\left(W^\top (x_n \circ r_i)\right) \circ s_i\right)
\end{aligned}
\quad \xrightarrow{\text{parallelize}} \quad Y = \phi\left(\left((X \circ R)W\right) \circ S\right)
$$




- **Rank 1 - BNNs (<u>ICML-20</u>):** learn rank-1 perturbation via variational inference, exploit hierarchical prior with non-centered parameterization

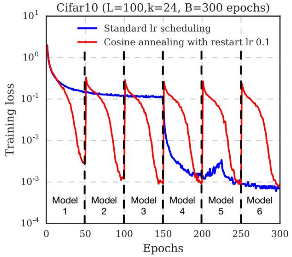
$$
\begin{aligned}
\mathcal{L} = &-\mathbb{E}_{q(r)q(s)} \log p(\mathcal{D}|W, r, s) \\
&+ \mathbb{KL}(q(r)\|p(r)) + \mathbb{KL}(q(s)\|p(s)) - \log p(W)
\end{aligned}
$$

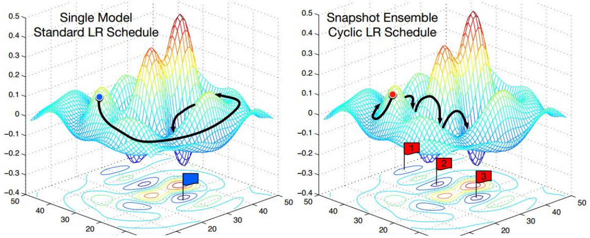


# The state-of-the-art and a unified view

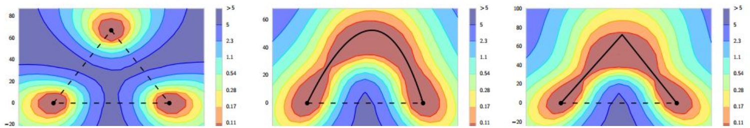
❖ **Several variants of deep ensemble:** *inspired by loss landscape*

- **Snapshot ensemble (ICLR-17)**: training SGD with **cyclicial** learning rate schedule --> train 1, get M for free



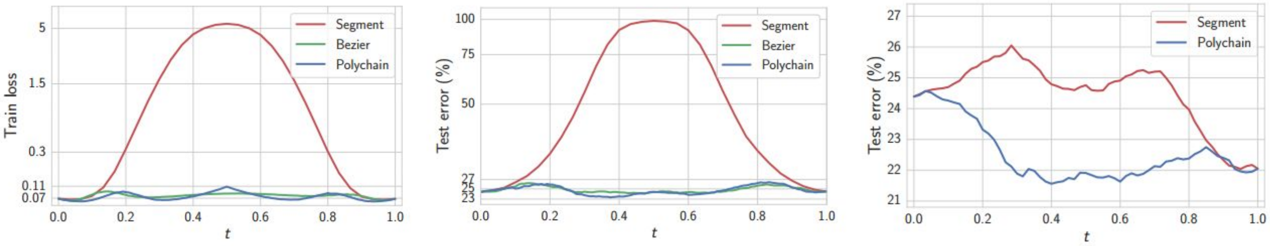




- **Fast Geometric Ensemble (NeurIPS-18)**: ensembling over **low-loss tunnel** connecting two minima --> cost of conventional training



**Polygonal chain:** $$\phi_\theta(t) = \begin{cases} 2\left(t\theta + (0.5-t)\hat{w}_1\right), & 0 \le t \le 0.5 \\ 2\left((t-0.5)\hat{w}_2 + (1-t)\theta\right), & 0.5 \le t \le 1. \end{cases}$$

**Bezier curve:** $$\phi_\theta(t) = (1-t)^2\hat{w}_1 + 2t(1-t)\theta + t^2\hat{w}_2, \quad 0 \le t \le 1.$$




** **high-performing** but **diverse** ensemble not need different minima.

# The state-of-the-art and a unified view

❖ **Bayesian model averaging:** unifying ensemble and Bayes

## Bayes vs Ensembles: What's the difference?

Both aggregate predictions over a collection of models. There are two core distinctions.

**The space of models.**

**Bayes** posits a prior that weighs different probability to different functions, and over an infinite collection of functions.

**Ensembles** weigh functions equally a priori and use a finite collection

**Model aggregation.**

**Bayesian** models apply averaging, weighted by the posterior.

**Ensembles** can apply any strategy and have non-probabilistic interpretations.

--> but all for **same goal**: to compute an accurate predictive distribution
--> do not need samples from a posterior, or even a faithful approximation to the posterior.

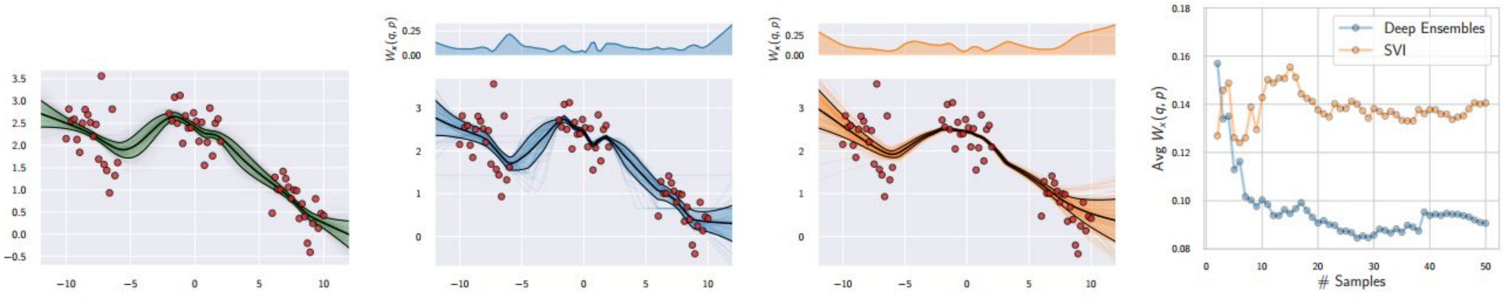
# The state-of-the-art and a unified view

❖ **Bayesian model averaging (NeurIPS-20):** unifying ensemble and Bayes

- derived from **marginalization procedure:** key distinguishing property of Bayesian method.
- an ensemble containing many **high-performing** but **diverse** models:

$$\boxed{p(y|x,\mathcal{D}) = \int p(y|x,w)p(w|\mathcal{D})dw}$$

  - consider the BMA integral *separately from* the simple Monte Carlo approximation in BNNs
  - Deep ensemble is non-Bayesian method, but can be treated as a compelling approach of BMA:




(a) Exact (b) Deep Ensembles (c) Variational Inference (d) Discrepancy with True BMA

# The state-of-the-art and a unified perspective

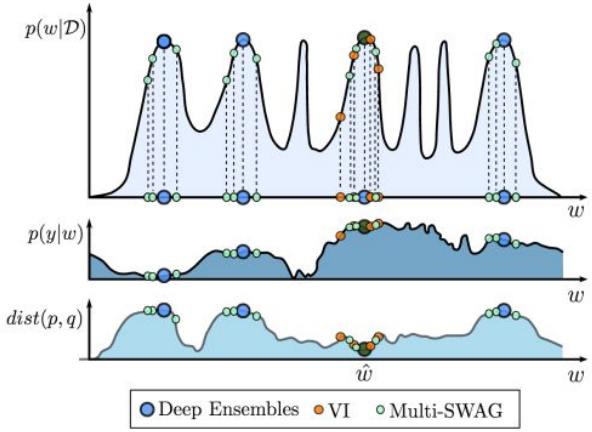
❖ **Bayesian model averaging:** unifying ensemble and Bayes

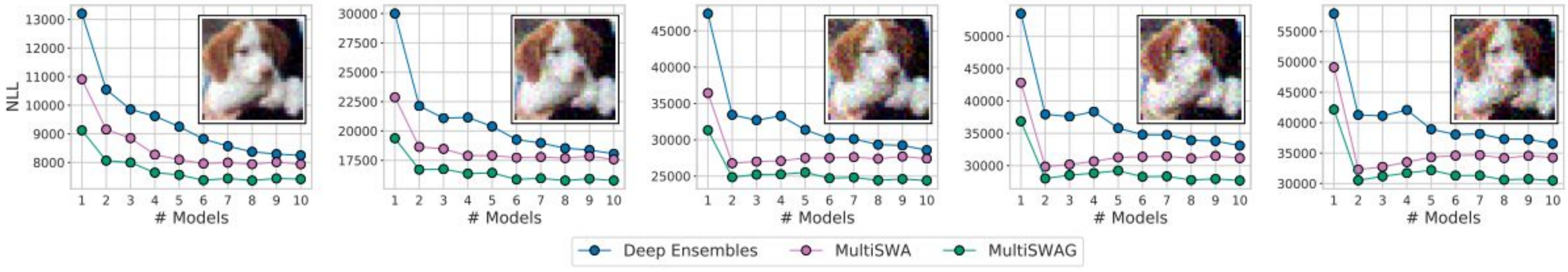
**Why BMA is actually compelling for deep learning ?**

- motivated by classical theory of statistical models
- evidenced by extensive empirical results
- provide **complementary benefits:**

*Ensemble MC-Dropout, Multi-SWAG, Multi-SWA*

*(Ensemble + local approximate/SWA can outperform Deep ensemble)*



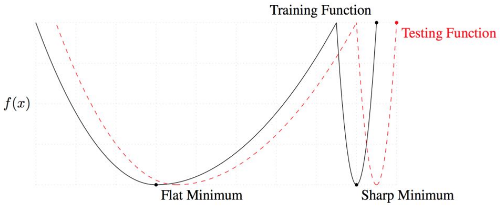

(a) Gaussian Noise

# The state-of-the-art and a unified perspective

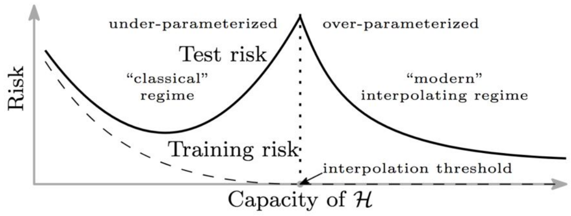
❖ **Bayesian model averaging:** unifying ensemble and Bayes

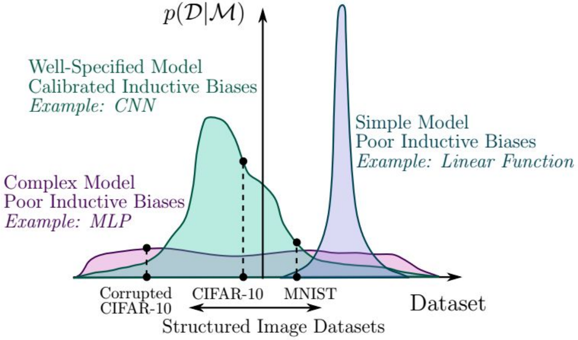
**Why BMA is actually compelling for deep learning ?**

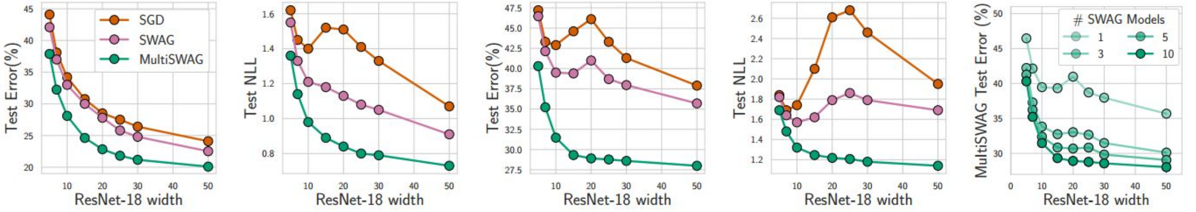
- provide intriguing perspectives on many problems of deep learning




Keskar et. al, ICLR 2017.
On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima.




(a) True Labels (Err) (b) True Labels (NLL) (c) Corrupted (Err) (d) Corrupted (NLL) (e) Corrupted (# Models)

# Content

## A. Uncertainty in Deep Learning

1. Background
2. Main approaches
3. The state-of-the-art and a unified perspective
4. Some potential research

# Some potential research

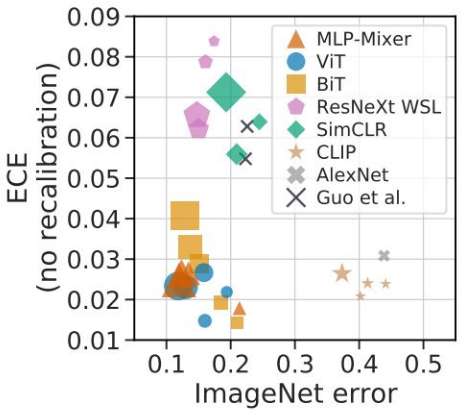
❖ **Some comments:**

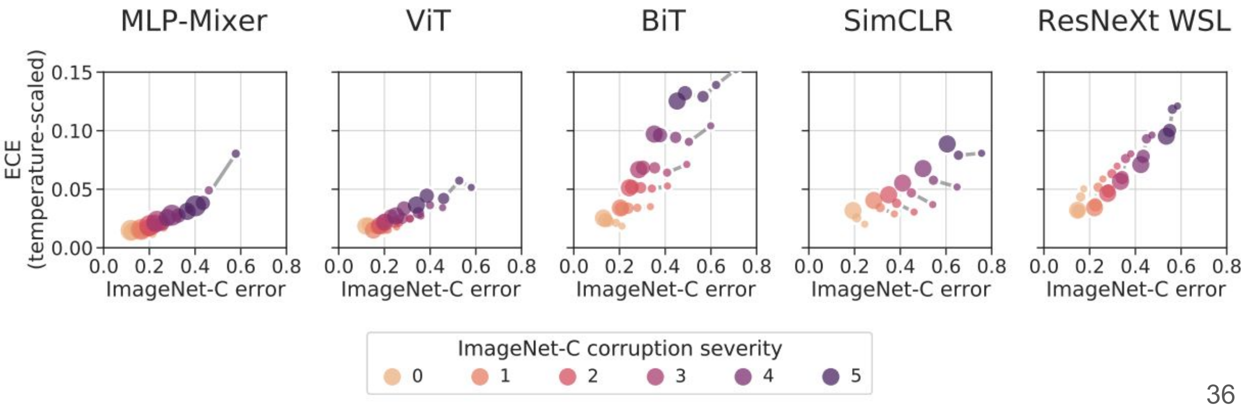
- robustness: improving model calibration under **distribution shift** is challenging, but prerequisite in practice
- subspace inference:
  - motivated by loss landscape characteristics
  - suggests integrating **Bayesian-like layers** into deep architectures.
- functional-space inference in BNNs
  - promising perspective to understand predictive distribution
  - avoid challenges of weight-space inference
  - connect to kernel methods, combine principles of Bayesian non-parametric with deep learning.

# Some potential research

❖ **Beyond principled approaches:** Stop thinking about just probability distributions. Leverage the inductive biases of core DL techniques --> improve significantly model calibration.

- test-time data augmentation
- mixup training: $x = \alpha x_1 + (1-\alpha)x_2, y = \alpha y_1 + (1-\alpha)y_2$
- more modern and more accurate architectures ([arXiv-21]): MLP-Mixer, Vision Transformer --> **reversed trends**
  - in-distribution: calibration slightly deteriorates with increasing model size
  - under distribution shift: accuracy and calibration are correlated, calibration improves with model size



# Content

## B. Bayesian neural network and its controversies

1. Why Bayesian neural nets
2. Expressive or simple approximate posterior distribution
3. Tempered or original true posterior distribution
4. Informative or vague prior distribution