Depth Uncertainty in Neural Networks

AMLab Seminar – 10/12/2020

James Allingham & Javier Antorán

Joint Work

Javier Antorán
ja666@cam.ac.uk

James Urquhart Allingham
jua23@cam.ac.uk

José Miguel Hernández-Lobato
jmh233@cam.ac.uk

Uncertainty in Deep Learning…

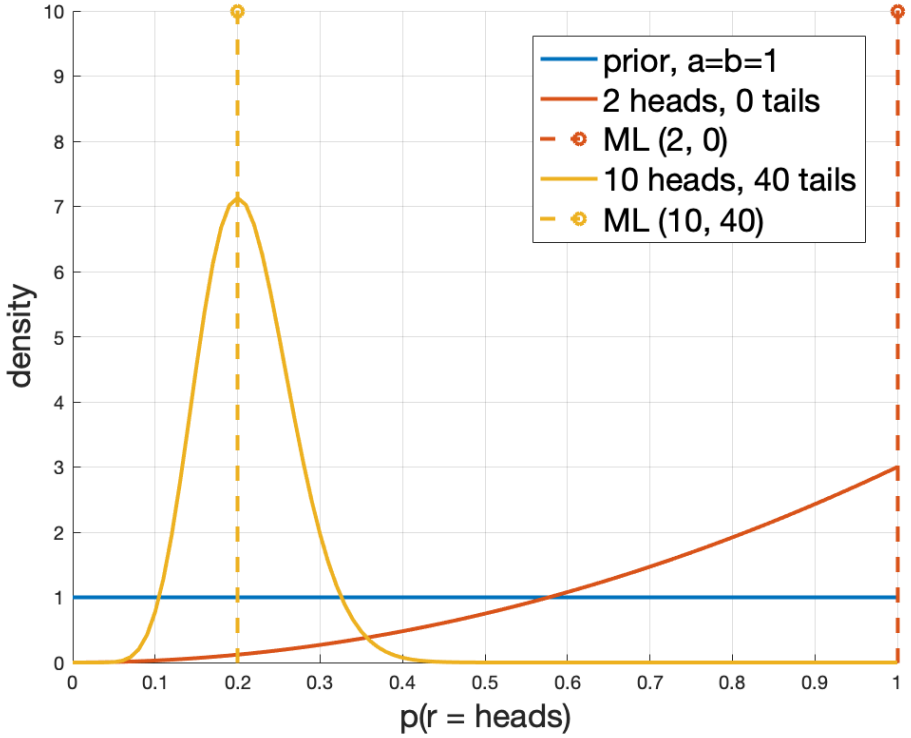
People saying AI will take over the world:

Meanwhile, my Deep Neural Network:

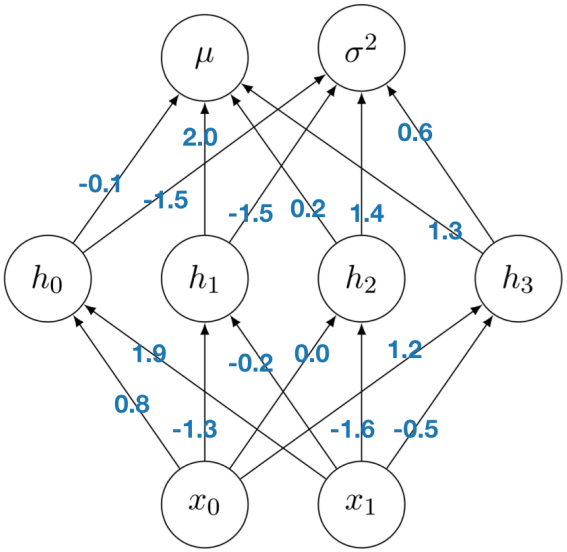
Probabilistic Inference: A biased coin

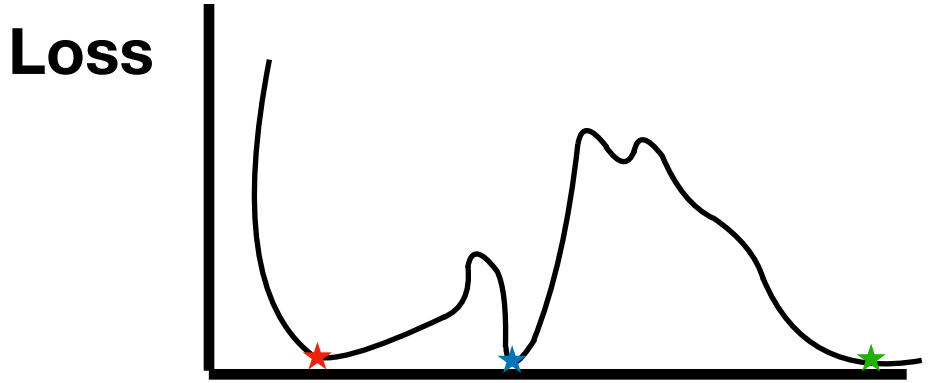
Likelihood Prior

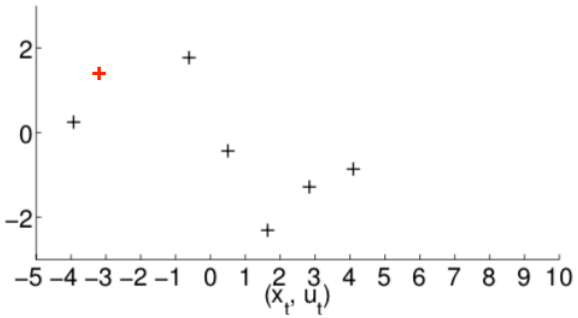
$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta) p(\theta)}{p(\mathcal{D})}$$

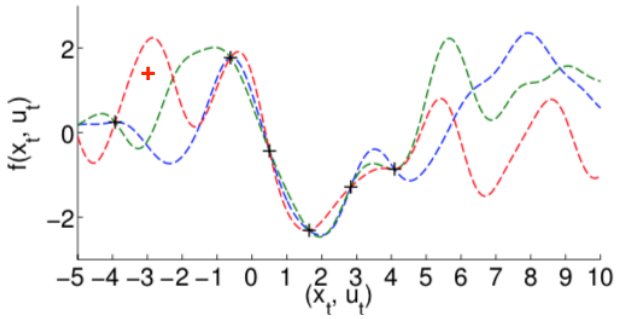


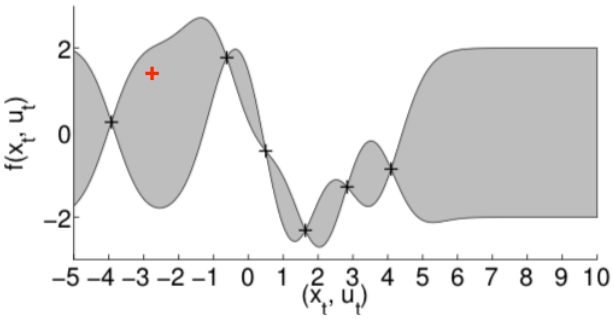
Different Weight Configurations yield Diverse Predictions



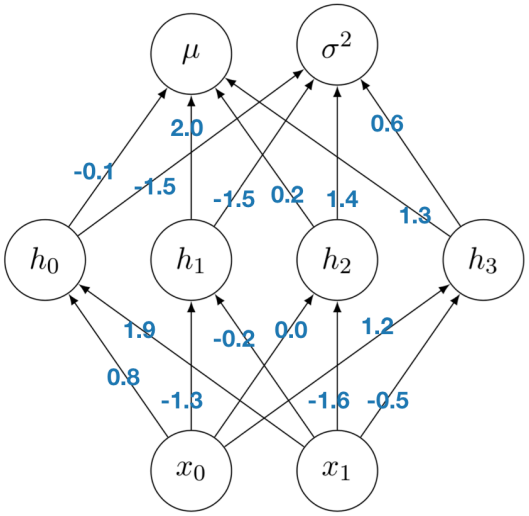


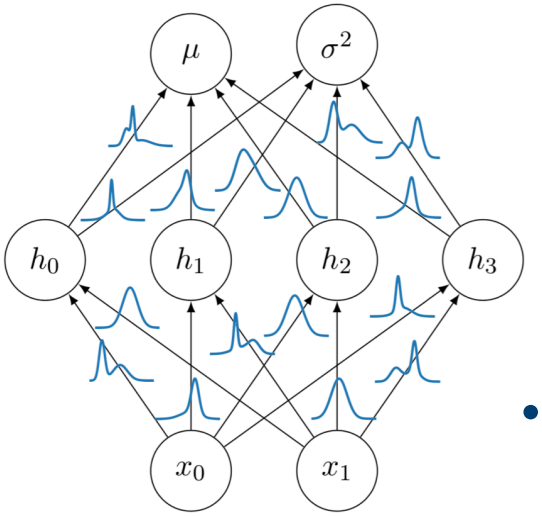






Weight Space Bayesian Neural Networks (BNNs)





- Predictions incorporate model uncertainty!

$$p(\mathbf{Y}^* \mid \mathbf{X}^*, \mathscr{D}) = \int p(\mathbf{Y}^* \mid \mathbf{X}^*, \theta) p(\theta \mid \mathscr{D}) d\theta$$

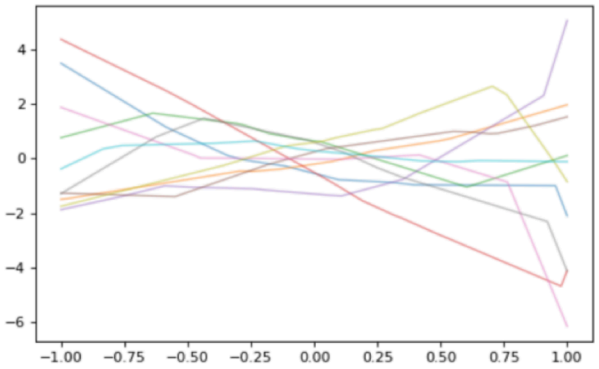
But Wait! There are 2 big issues:

How to Specify Meaningful Priors?

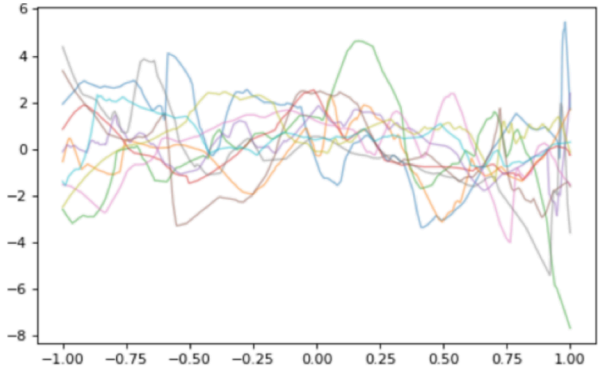
Samples from fully connected BNN prior converge to white noise!

$$p(\theta) = \mathcal{N}(\theta; 0, I)$$

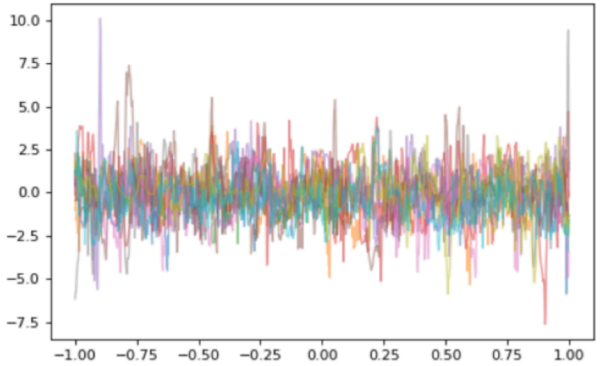
1 Hidden Layer



5 Hidden Layer

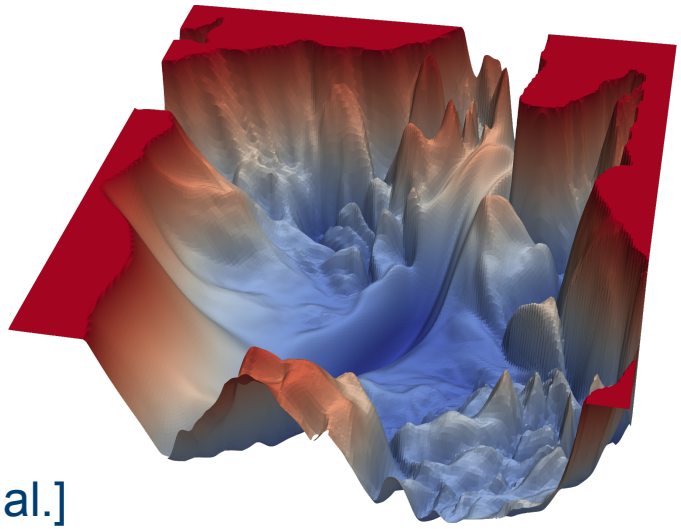


20 Hidden Layer



The Weight Posterior is Intractable

- The data's likelihood under a BNN model is a very complex, high dimensional and multimodal function.



[Li et al.]

- **Intractable evidence:**

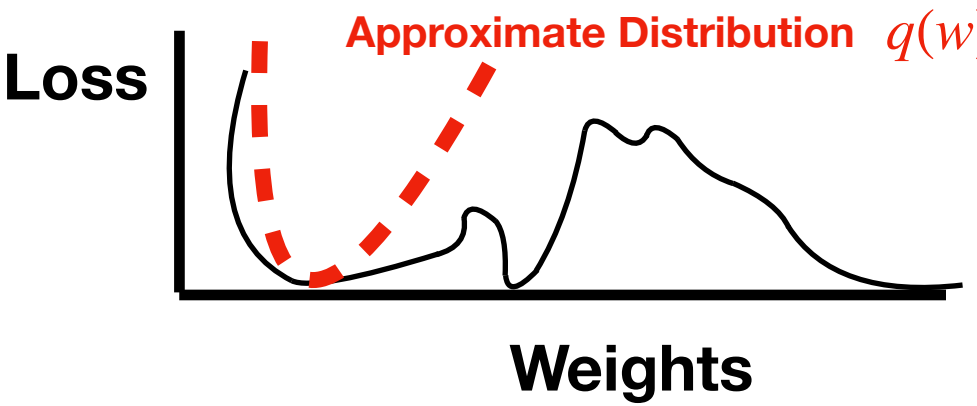
$$p(\mathscr{D}) = \int p(\mathscr{D} \mid \theta) p(\theta) d\theta$$

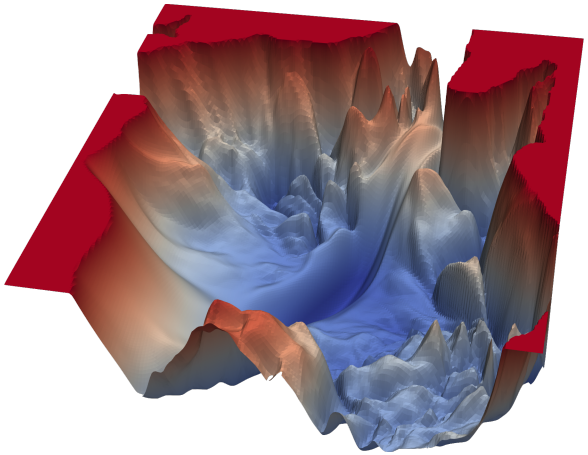
- **Intractable predictive:**

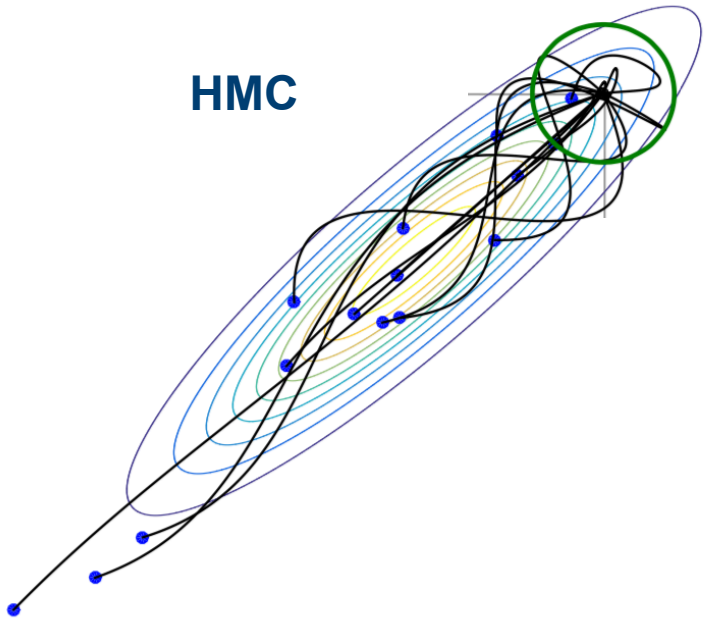
$$p(\mathbf{Y}^* \mid \mathbf{X}^*, \mathscr{D}) = \int p(\mathbf{Y}^* \mid \mathbf{X}^*, \theta) p(\theta \mid \mathscr{D}) d\theta$$

- Must resort to **Approximate Inference**

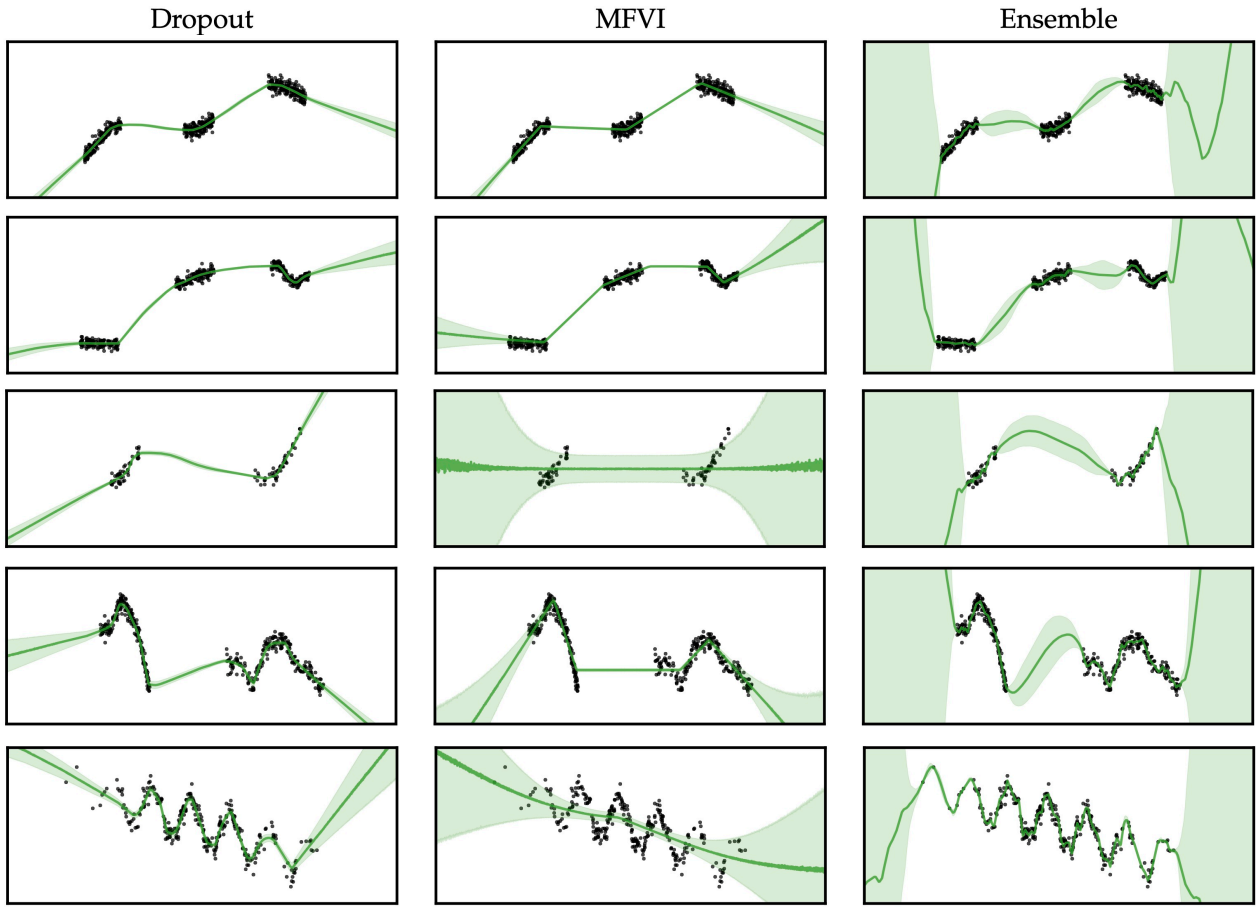
Our Uncertainty Estimates are Almost Always Biased by Our Approximations







As a result: Unreliable Predictions and Uncertainty Estimates



Detailed Studies:

Foong et al. "**On the expressiveness of approximate inference in bayesian neural networks.**" *NeurIPS* (2020).

Wenzel et al. "**How Good is the Bayes Posterior in Deep Neural Networks Really?**" CoRR abs/2002.02405 (2020)

Try for yourself:

github.com/JavierAntoran/Bayesian-Neural-Networks

Can we Elude the Difficulties of Weight Space?

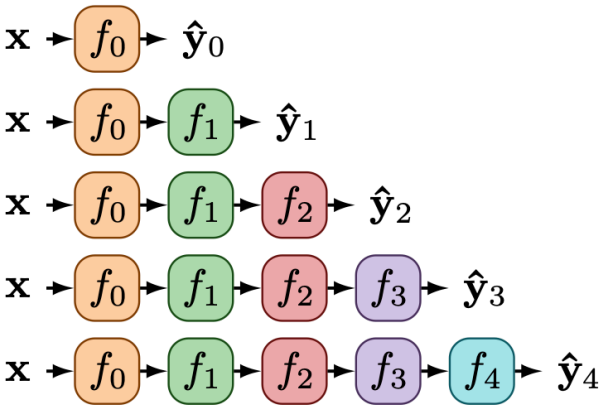
- Weight space inference is difficult: $$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})}$$

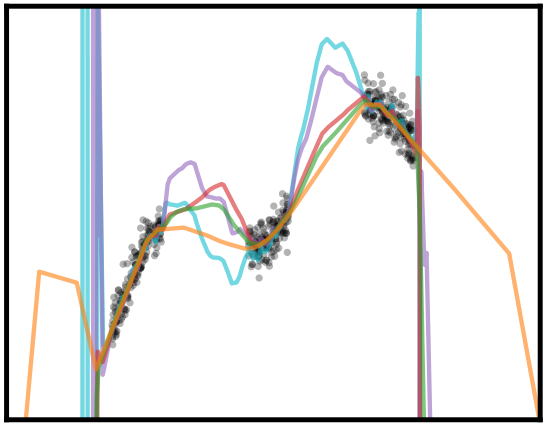
- We can generalise by explicitly considering our model class: $$p(\theta \mid \mathcal{D}, \mathcal{M}) = \frac{p(\mathcal{D} \mid \theta, \mathcal{M})p(\theta \mid \mathcal{M})}{p(\mathcal{D} \mid \mathcal{M})}$$

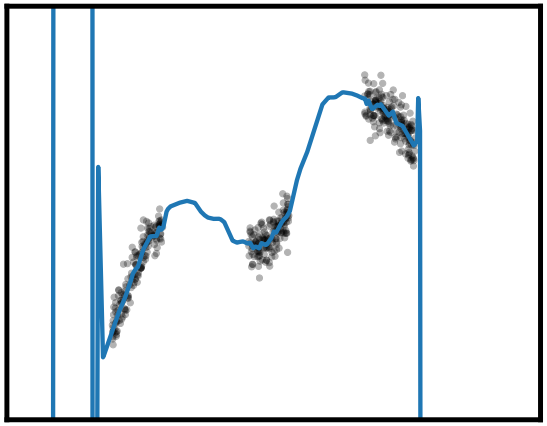
- Neural Architecture Search performs MAP inference over $\mathcal{M}$: $$p(\mathcal{M} \mid \mathcal{D}; \theta) \approx \delta(\mathcal{M} - \mathcal{M}^*);$$ $$\mathcal{M}^* = \text{argmax}_{\mathcal{M}} p(\mathcal{D} \mid \mathcal{M}; \theta)p(\mathcal{M})$$

- We perform full inference over $\mathcal{M}$: $p(\mathcal{M} \mid \mathcal{D}; \theta) = \frac{p(\mathcal{D} \mid \mathcal{M}; \theta)p(\mathcal{M})}{p(\mathcal{D}; \theta)}$

DUN: Intuition

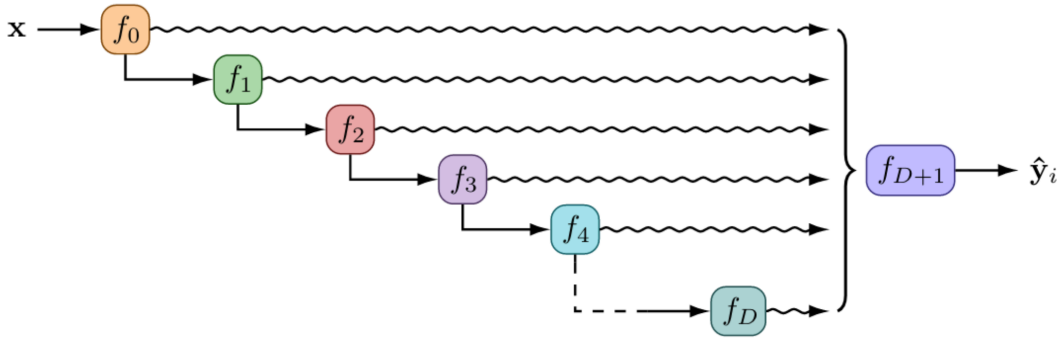


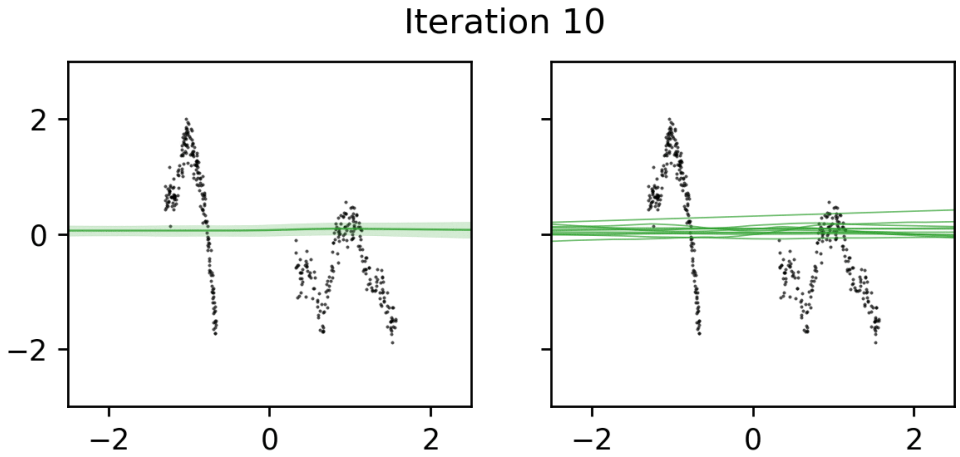




- Depth Uncertainty Networks (DUNs), transform uncertainty over depth into predictive uncertainty.

It's not Intractable if its countable! (In this case)

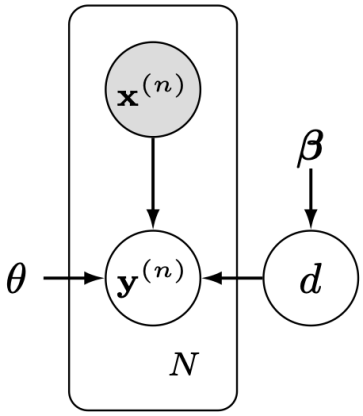


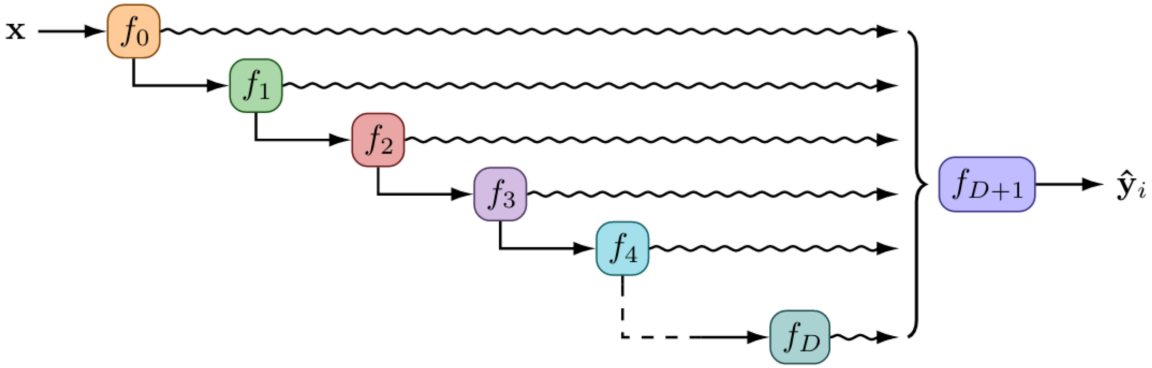


- We perform inference over network depth (discrete space)
- Sequential computation is amortised!

$$p(y = c \,|\, \mathbf{x}^{(n)}, d = i; \theta)$$
$$= f_{D+1} \circ (f_i \circ f_{i-1} \ldots \circ f_1)(\mathbf{x}^{(n)})$$

DUN: Inference with a Single Forward Pass





$$d \sim \text{Cat}(d; \beta)$$

$$\log p(\mathfrak{D}; \boldsymbol{\theta}) = \log \sum_{i=0}^{D} \left(p_{\boldsymbol{\beta}}(d{=}i) \cdot \prod_{n=1}^{N} p(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, d{=}i; \boldsymbol{\theta}) \right)$$

$$\geq \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \mathbb{E}_{q_{\boldsymbol{\alpha}}(d)} \left[\log p(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, d; \boldsymbol{\theta}) \right] - \text{KL}(q_{\boldsymbol{\alpha}}(d) \,\|\, p_{\boldsymbol{\beta}}(d))$$

But Why VI? → Optimising Hyperparameters (θ) is Hard

- With the ML objective, the rich get richer and the posterior is prone to collapse

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathfrak{D}; \boldsymbol{\theta}) = \sum_{i=0}^{D} p(d{=}i|\mathfrak{D}; \boldsymbol{\theta}) \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathfrak{D}|d{=}i; \boldsymbol{\theta})$$

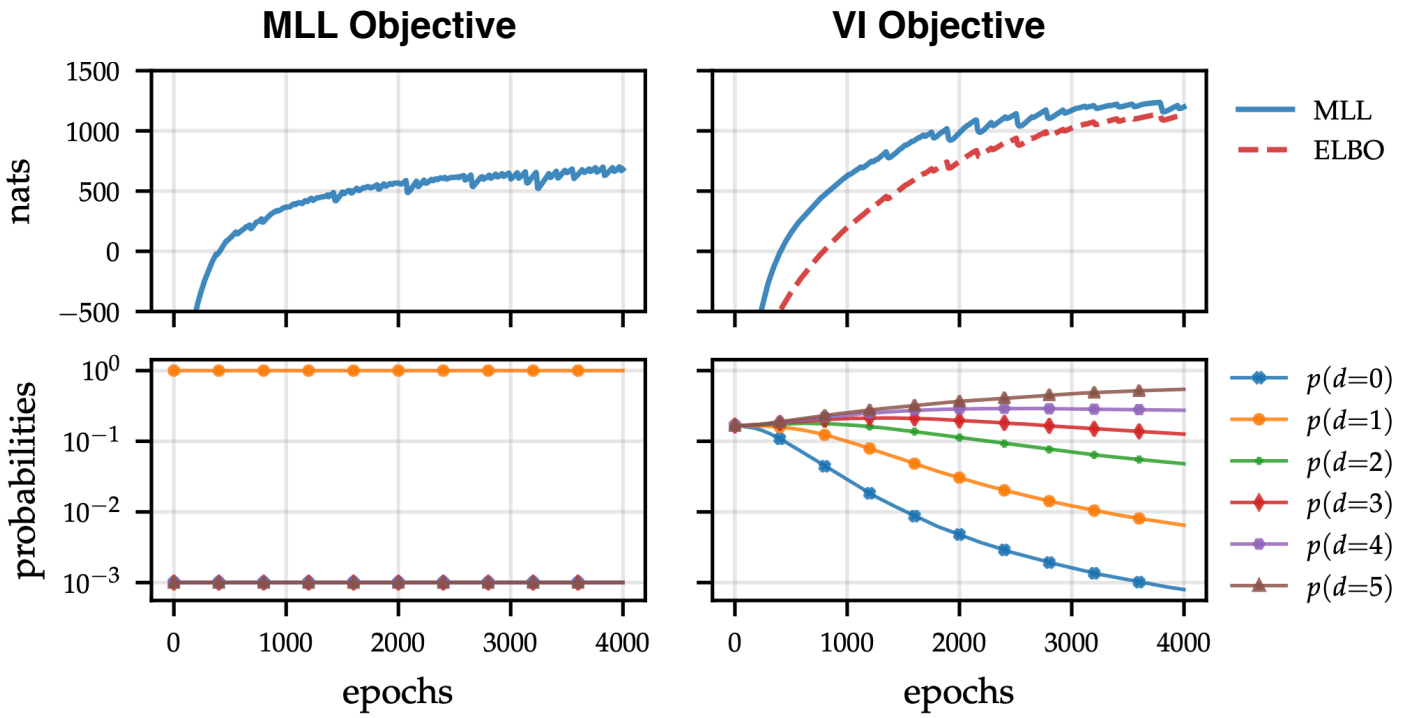
- With VI, the optimization of variational parameters and model weights is decoupled

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \sum_{i=0}^{D} q_{\boldsymbol{\alpha}}(d{=}i) \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathfrak{D}|d{=}i; \boldsymbol{\theta})$$

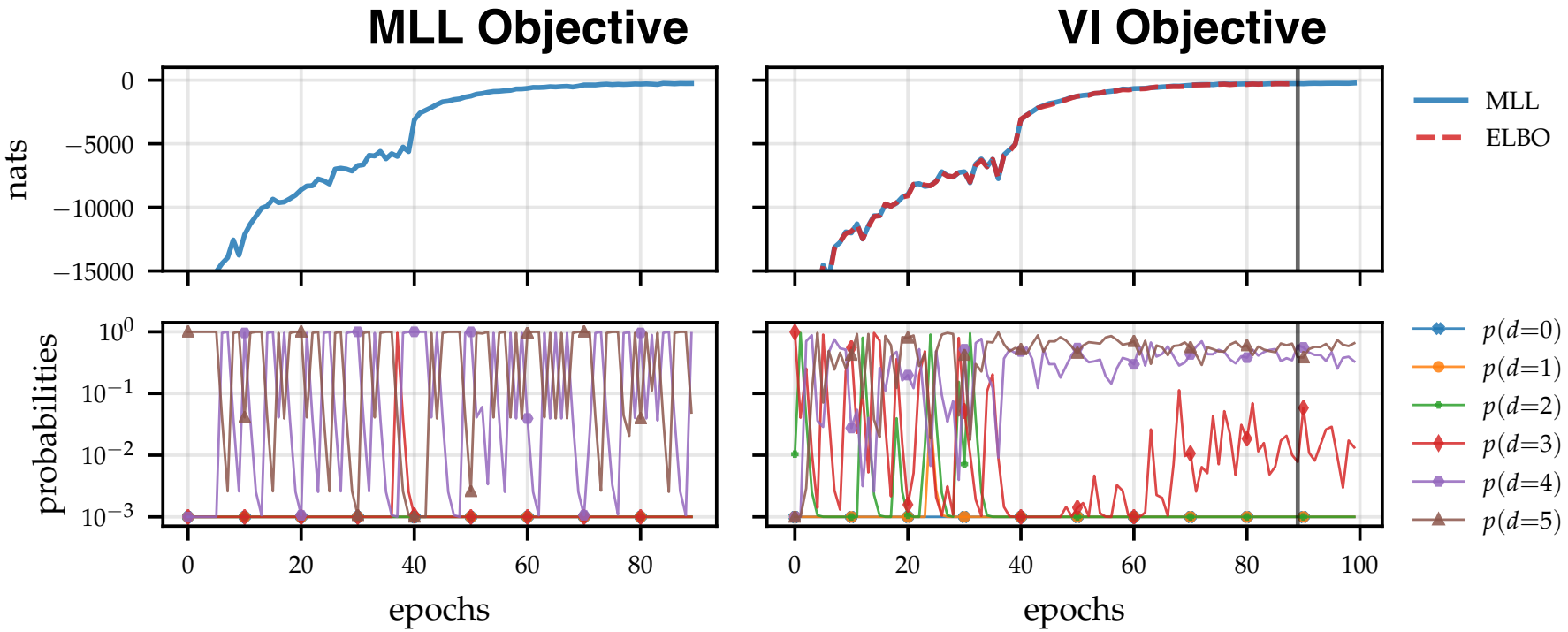
$$\frac{\partial}{\partial \alpha_i} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \log p(\mathfrak{D}|d{=}i; \boldsymbol{\theta}) \frac{\partial}{\partial \alpha_i} q_{\boldsymbol{\alpha}}(d{=}i) - (\log q_{\boldsymbol{\alpha}}(d{=}i) - \log p(d{=}i) + 1) \frac{\partial}{\partial \alpha_i} q_{\boldsymbol{\alpha}}(d{=}i)$$

Posterior is updated with log likelihood instead of being ∝ likelihood

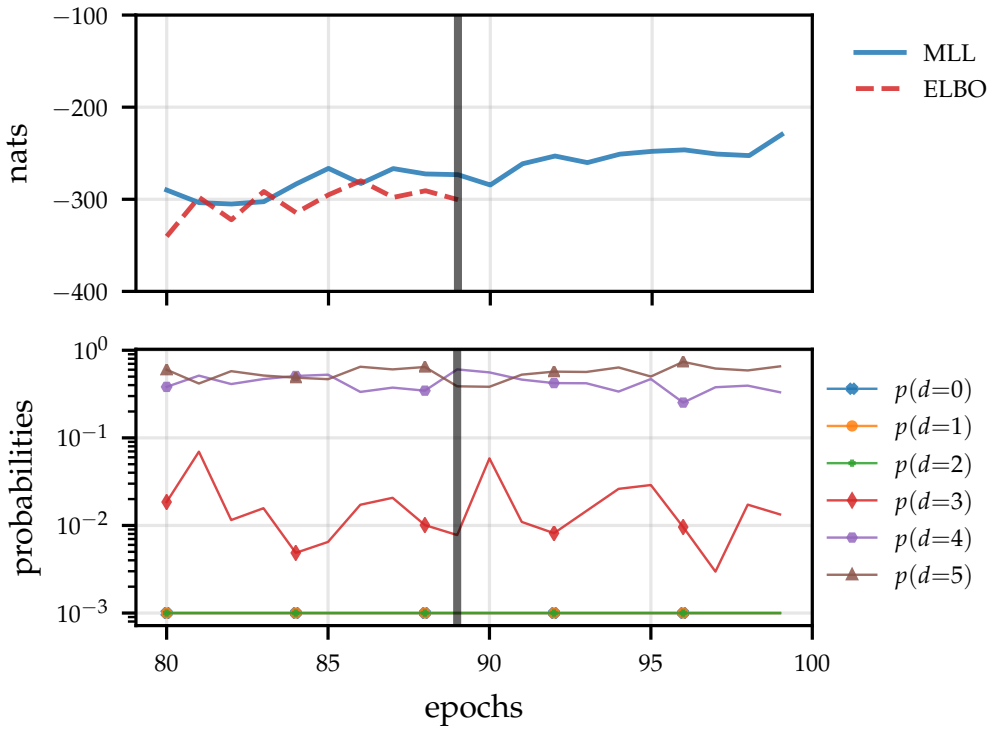
5 layer MLP on Concrete



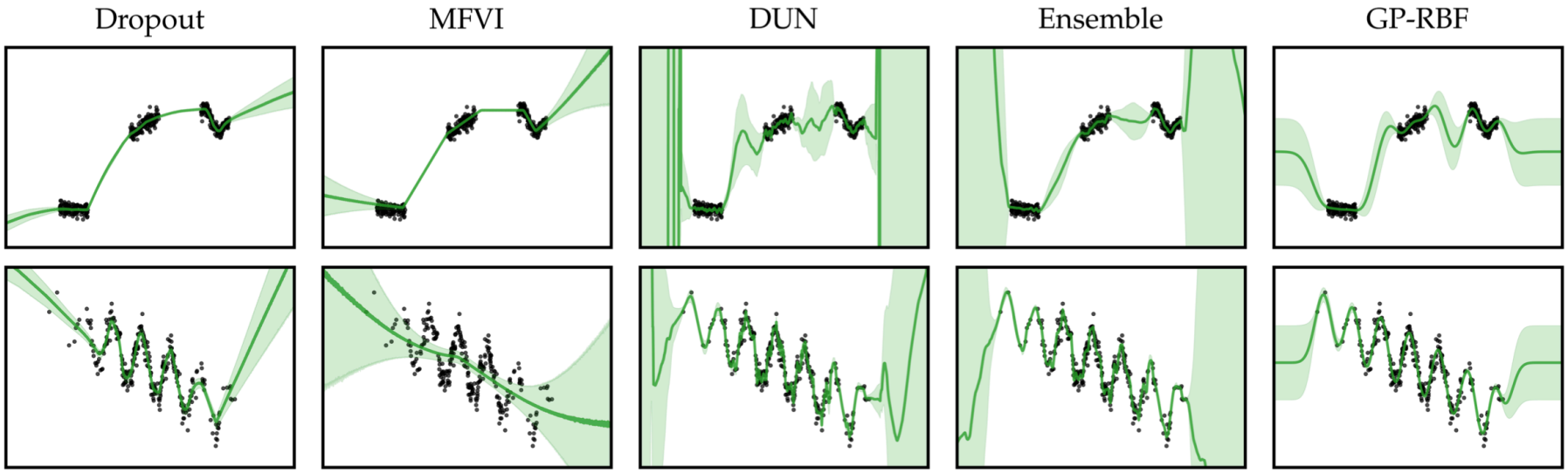
ResNet-50 on Fashion MNIST



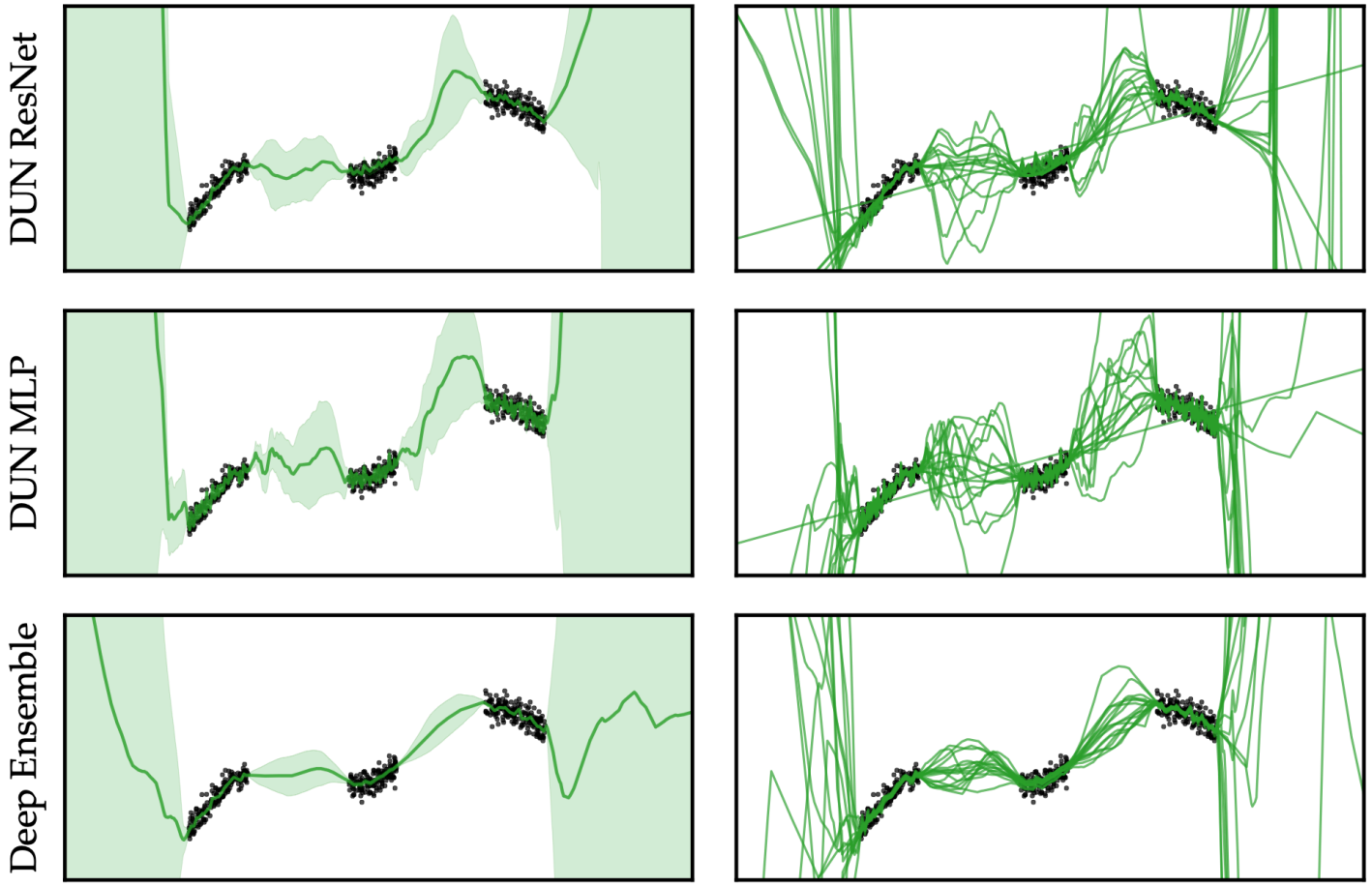
Enhance!


−100
−200
−300
−400
nats
MLL
ELBO
10^0
10^{-1}
10^{-2}
10^{-3}
probabilities
$p(d{=}0)$
$p(d{=}1)$
$p(d{=}2)$
$p(d{=}3)$
$p(d{=}4)$
$p(d{=}5)$
80
85
90
95
100
epochs

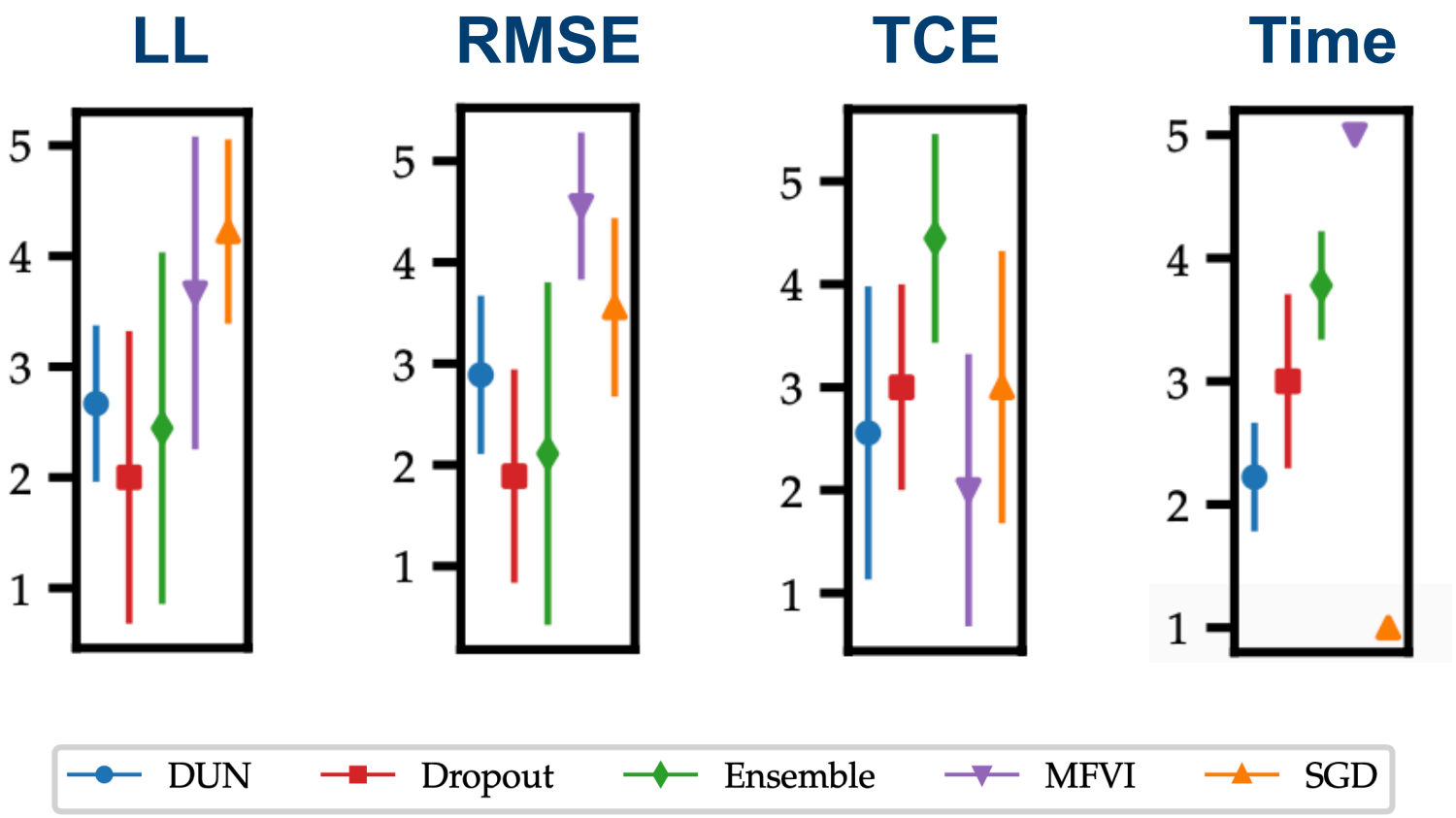
Toy Examples!



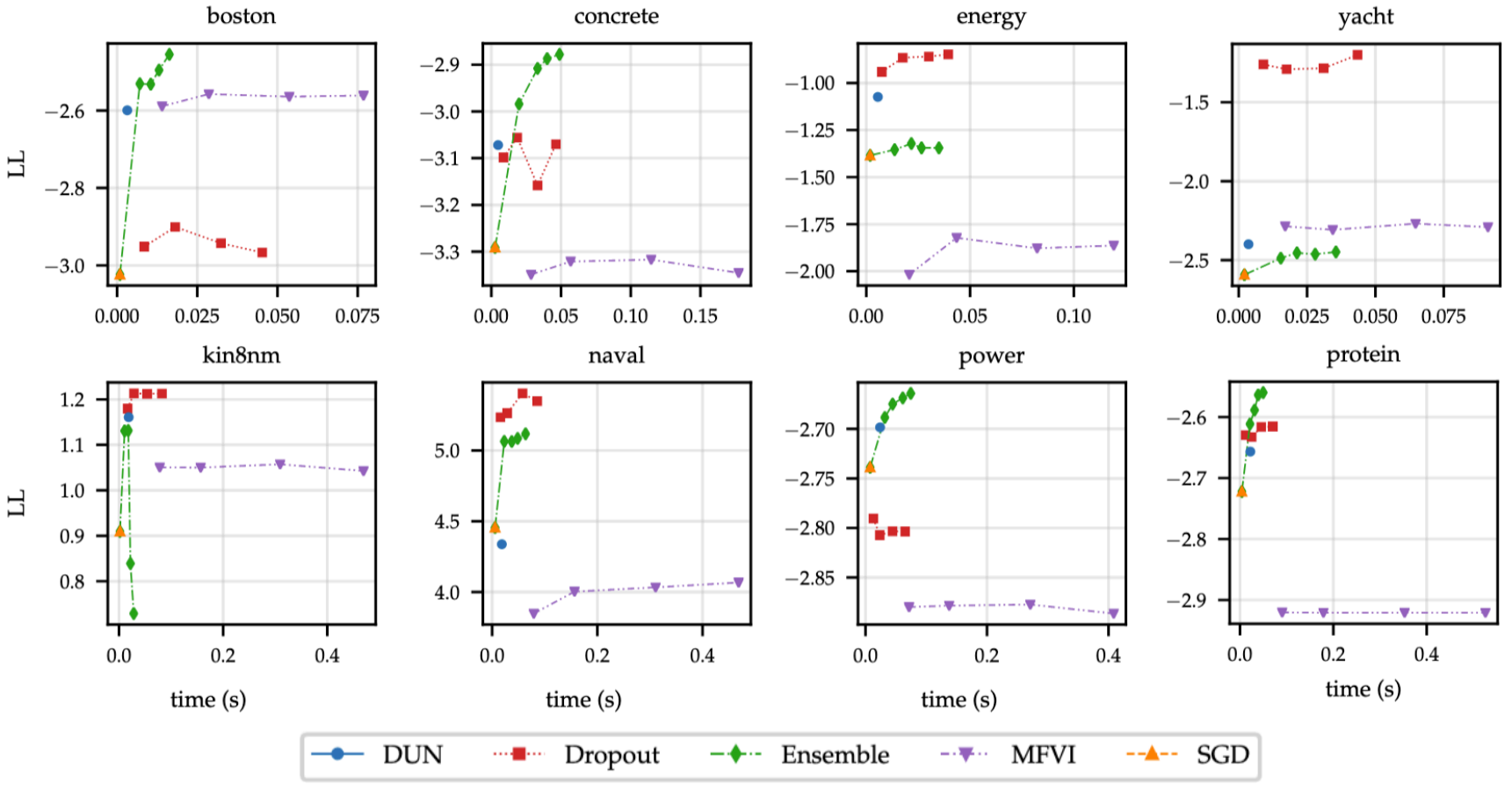
Function Diversity in DUNs vs Ensembles



Regression (ranks across 9 UCI datasets)



DUNs are Compute Efficient – UCI Edition



What if NN shape is not constant? →Adaption Layers

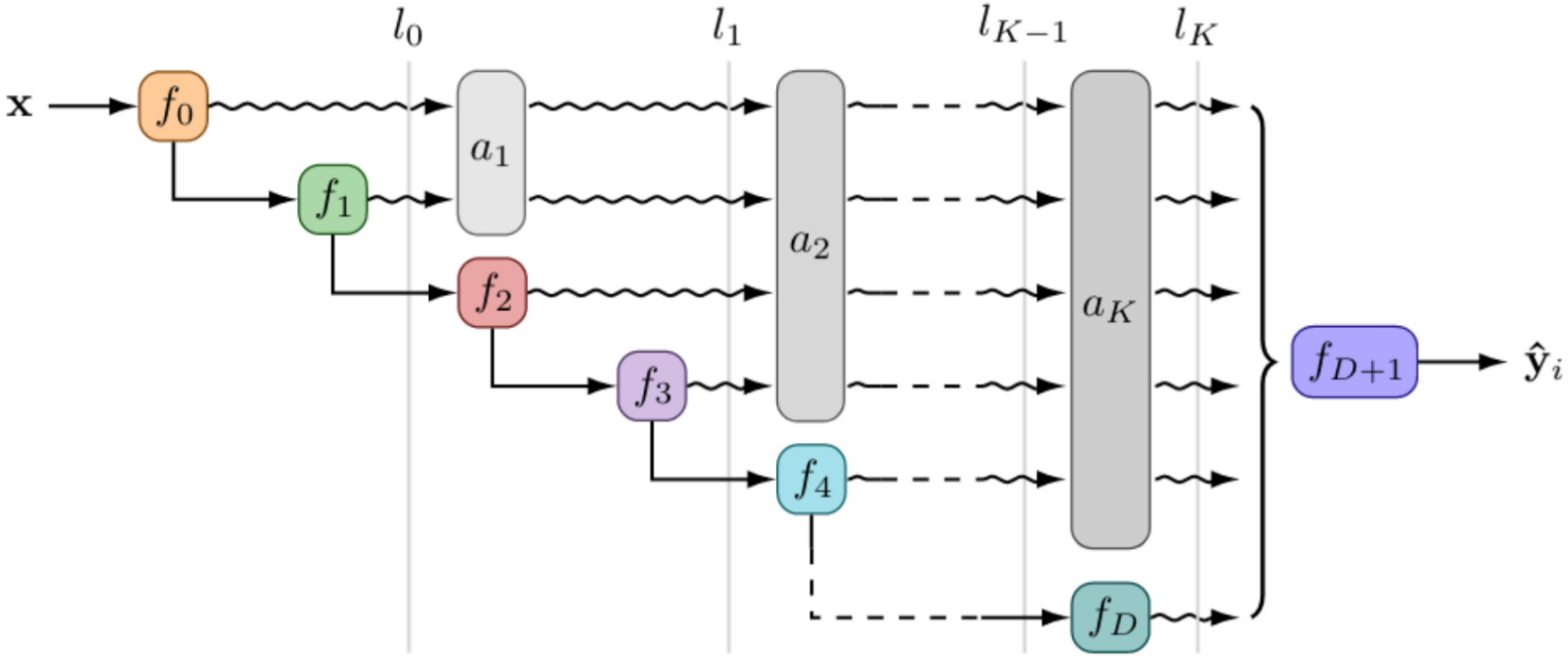
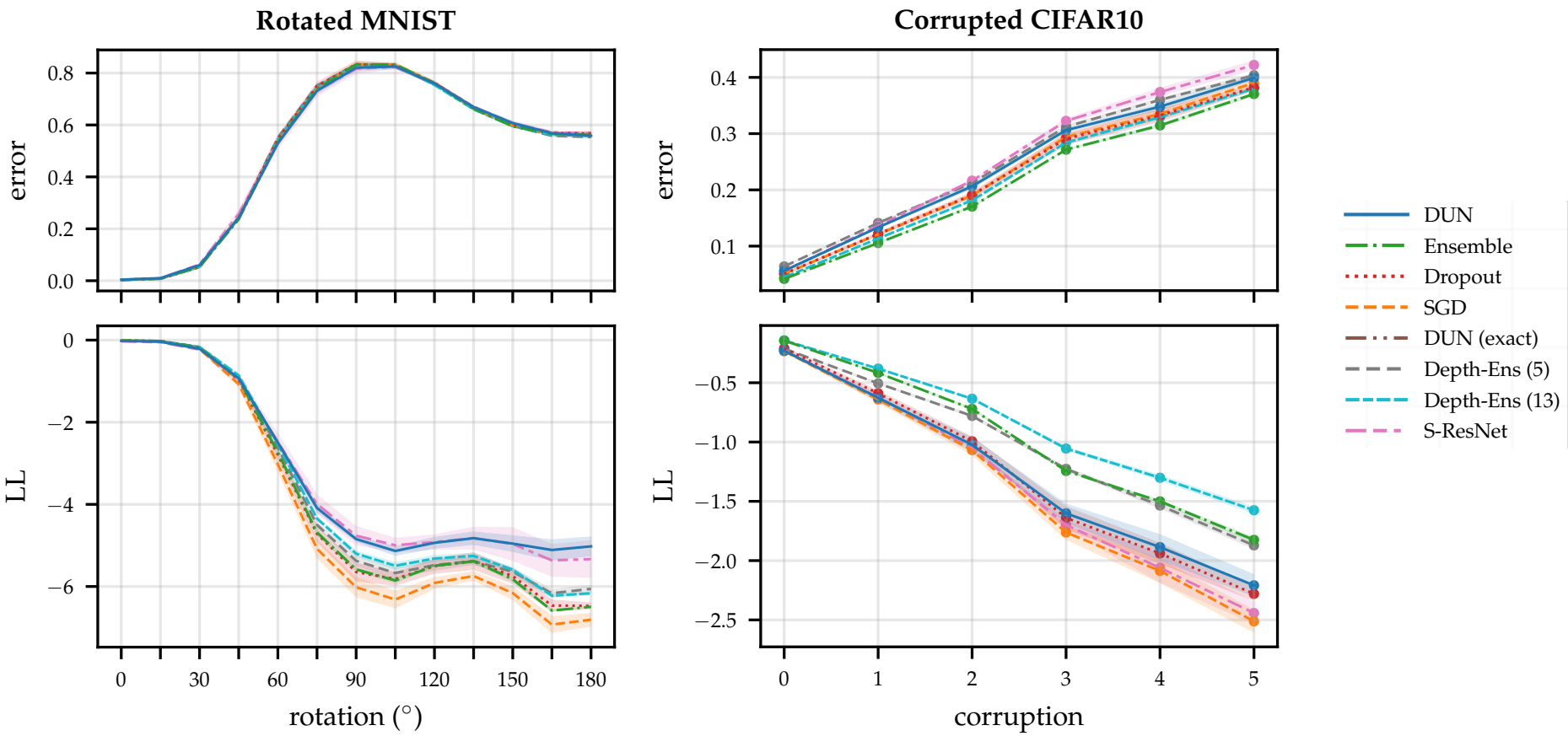
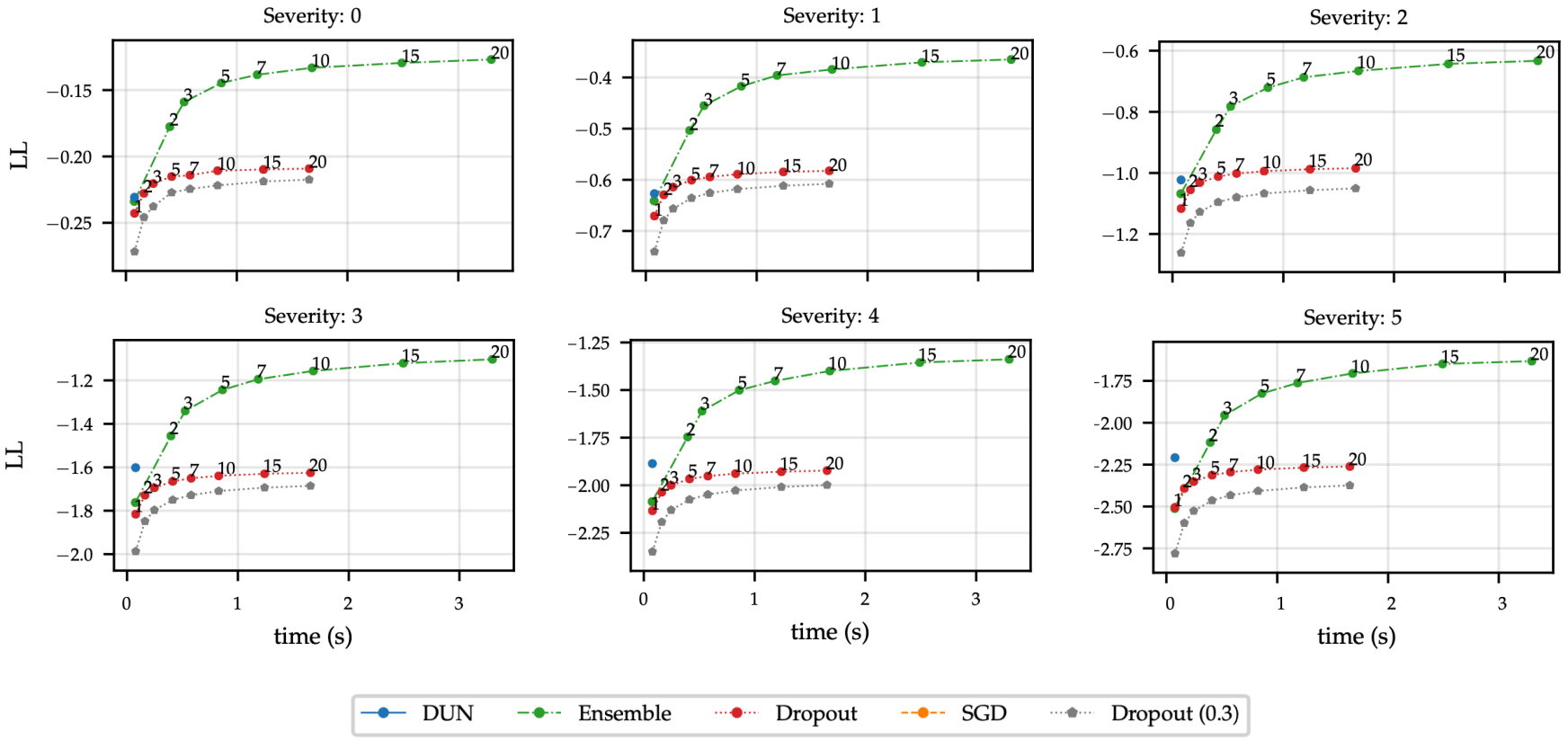


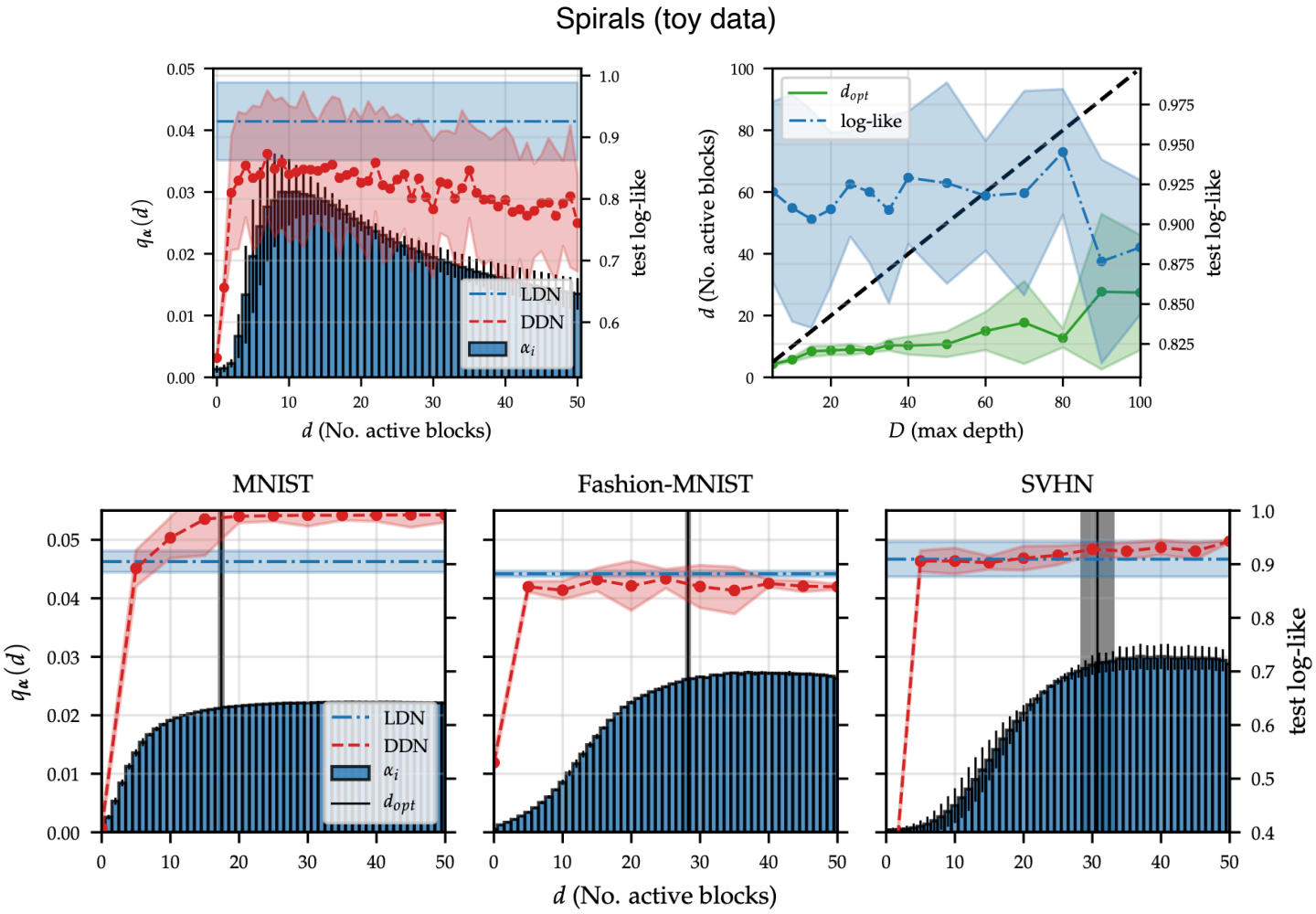
Image Classification (ResNet50)



DUNs are Compute Efficient – CIFAR10 Edition



Architecture Search with DUNs



Connections to other work

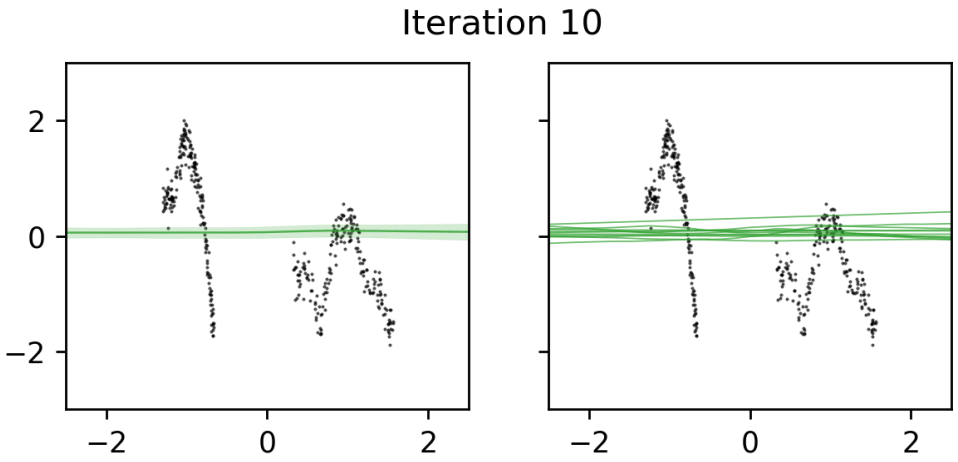
- Over-parameterised NNs are capable of representing multiple (diverse) models of the data
  - MIMO[1]
- Diversity in predictions gives good robustness
  - Hyper-deep ensembles[2]

[1]Havasi, Marton, et al. "Training independent subnetworks for robust prediction." *arXiv preprint arXiv:2010.06610* (2020).

[2]Wenzel, Florian, et al. "Hyperparameter ensembles for robustness and uncertainty quantification." *Advances in Neural Information Processing Systems* 33 (2020).

Summary

- Existing methods for estimating uncertainty in deep learning are computationally expensive and often perform poorly.
- Depth Uncertainty Networks (DUNs), transform uncertainty over depth into predictive uncertainty in a single forward pass.
- DUNs provide the best robustness vs compute time trade-off in both classification and regression with modern architectures.



Code

github.com/cambridge-mlg/DUN