On Bayesian Deep Learning

An Outsider's View

Jialin Lu, Feb 26 2020
Meeting of Ester Lab

SFU
SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

Why I am talking about this?

Well, during the NeurIPS 2019 conference..

There is a tutorial on Bayesian Deep Learning (which I actually failed to attend).

And then someone says this thing is kind of cool.

then I was somehow *volunteered* for this presentation

Outline of This talk

PART 1: Combining Bayesian and Deep Learning

Give the motivation

PART 2: Bayesian treatment of Deep Learning

Two technical approach.

PART 3: What goes wrong?

Why Part 2 is not working great compared with a simple baseline

PART 4: A New Hope

Advice and opinion (personal idea)

PART 1

Combining two approaches

Bayesian Learning and Deep Learning

Bayesian Learning

A general framework of modelling.

models uncertainty

Very flexible: adaptive setting, online learning.

Scale it to large datasets is hard.

Deep Learning

Another general framework of modelling.

in general does not model uncertainty

Very flexible (in different ways), design of architectures, ...

Can scale well to very large datasets.

BL and DL are good and bad in different ways

Can we combine these two?

Bayesian learning

Bayesian models
(GPs, BayesNets, PGMs,)

Bayesian inference
(Bayes rule)

Deep learning

Deep models
(MLP, CNN, RNN etc.)

Stochastic training
(SGD, RMSprop, Adam)

	Bayes	DL
Can handle large data and complex models?	✗	✓
Scalable training?	✗	✓
Can estimate uncertainty?	✓	✗
Can perform sequential / active /online / incremental learning?	✓	✗

image source, Emtiyaz Khan 2019

To be or not to be is not the question;
the vital question is how to be or how not to.

– *Abraham Joshua Heschel*

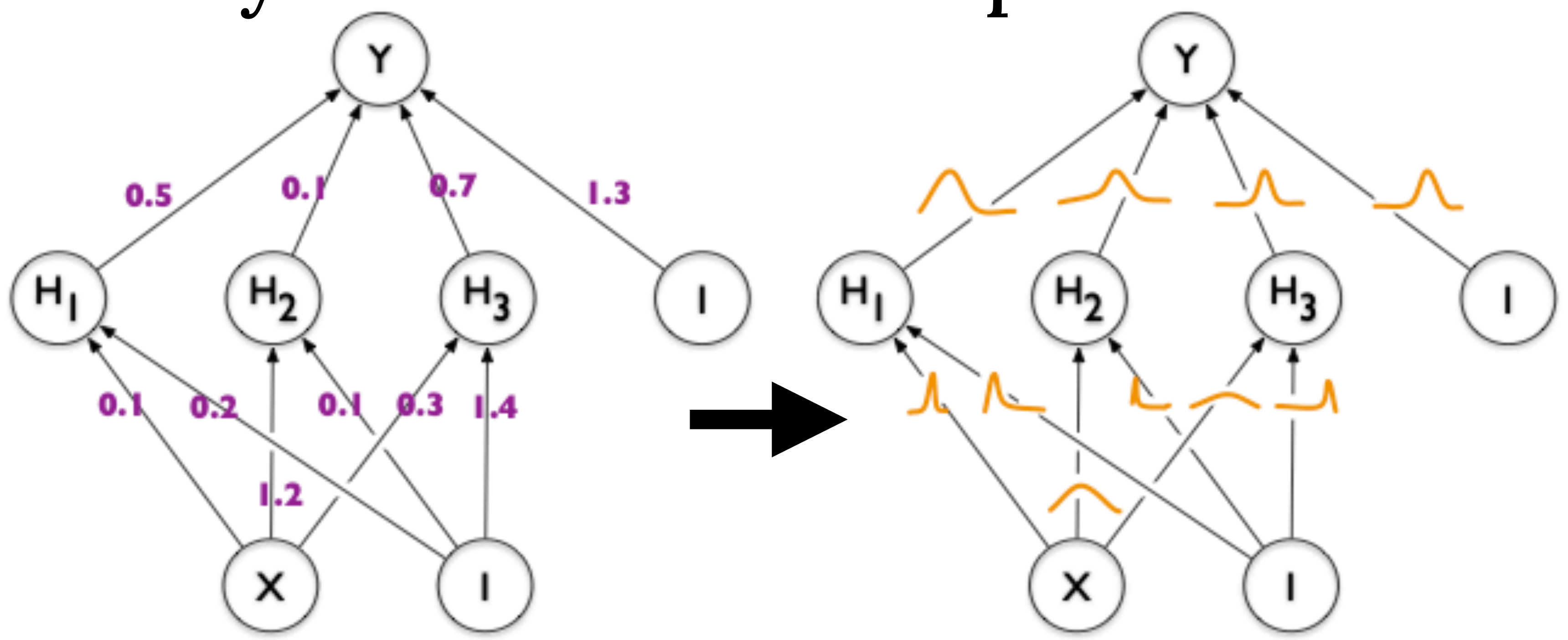
Two ways of combining
Deep Learning (DL) and Bayesian Learning (BL)

- DL->BL
  - Given a probabilistic graphical model, Deep learning can be used to model some directed edge.
  - For example: Variational auto encoder

- BL-> DL

Bayesian Deep Learning, Bayesian Neural Network

  - Consider to have uncertainty in the learned parameters of a Deep Neural Network.



PART 2

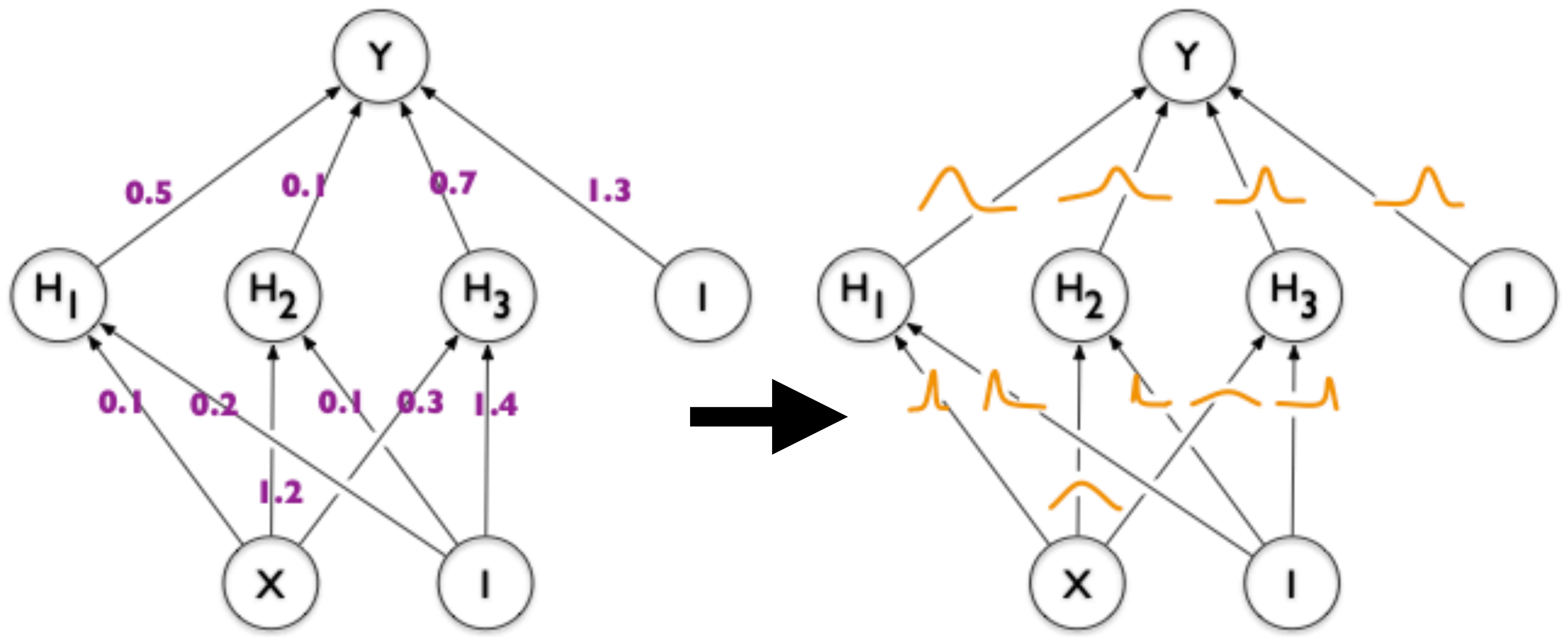
Bayesian Treatment of DL

i.e. Bayesian Neural Networks or Bayesian DL

Our goal is to obtain the posterior

In Bayesian Deep Learning, we wish to obtain a posterior distribution of the weights of the neural network, given some data.

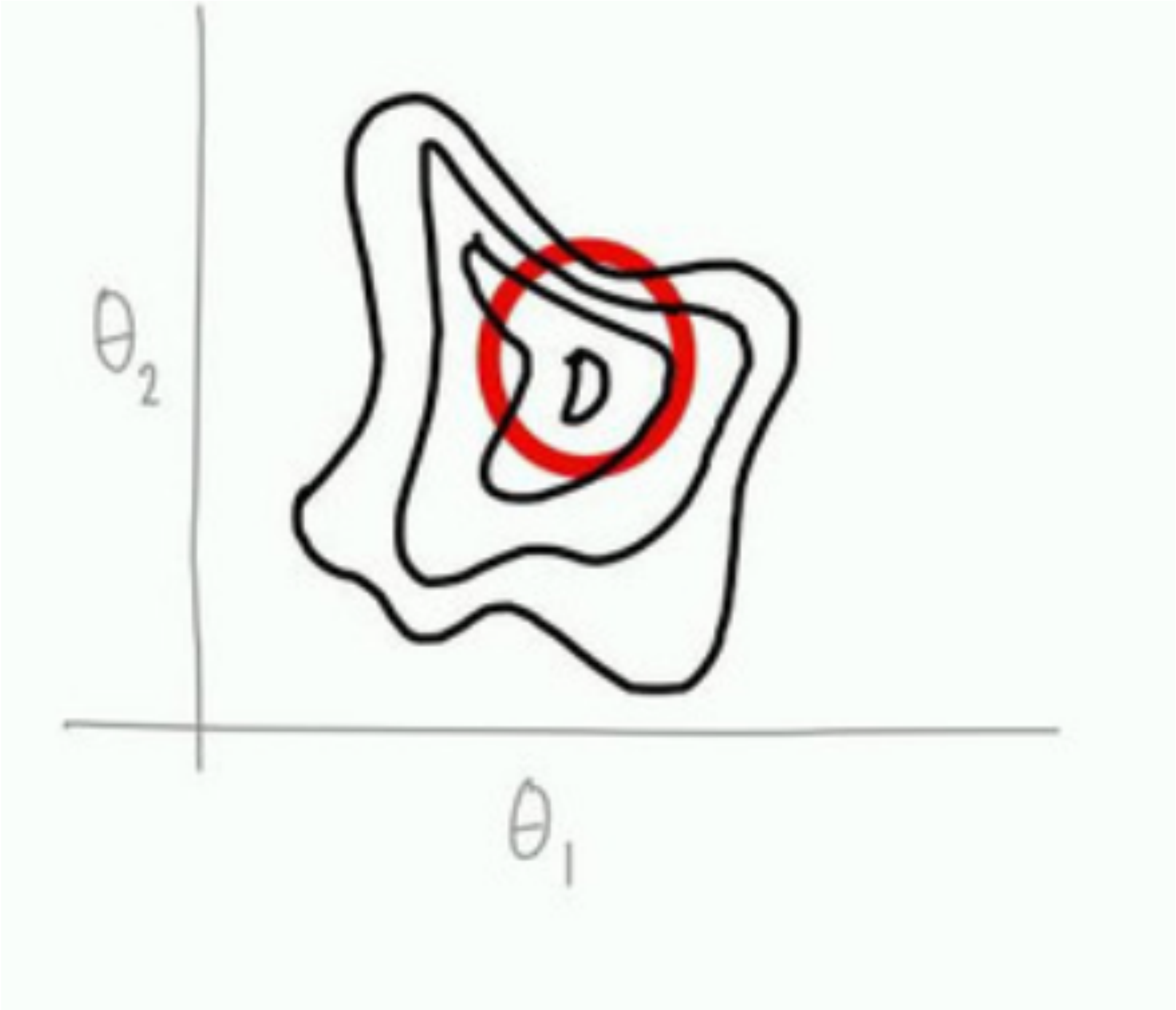
We call it posterior because it is computed by considering both the likelihood (how well it fits the data) and prior (how we favour certain models)



$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$$

Option 1: Variational Inference-based

We predefine the form of the posterior q, and adjust the parameters of q to approximate the true posterior



The NeurIPS tutorial
Optimization from a Bayesian view

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta} \quad \boxed{\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta)}$$

$$= \arg\min_{q \in \mathcal{P}} \; \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

All distribution Distribution Entropy

We now restrict *P* to *Q:* this is known as variational inference

$$\arg\min_{q \in \mathcal{Q}} \; \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

A unified framework:
Allows you to derive DL optimizer by choosing the assumption

$$\min_{\theta} \ \ell(\theta) \quad \text{vs} \quad \min_{q \in \mathcal{Q}} \ \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

Entropy

Deep Learning algo: $\theta \leftarrow \theta - \rho H_{\theta}^{-1} \nabla_{\theta} \ell(\theta)$

Bayes learning rule: $\lambda \leftarrow \lambda - \rho \nabla_{\mu} \left(\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q) \right)$

Natural and Expectation parameters of an exponential family distribution q

By setting to a fixed-variance Gaussian, we get SGD

Gaussian distribution	$q(\theta) := \mathcal{N}(m, 1)$
Natural parameters	$\lambda := m$
Expectation parameters	$\mu := \mathbb{E}_q[\theta] = m$
Entropy	$\mathcal{H}(q) := \log(2\pi)/2$

By setting to a fixed-variance Gaussian, we get Newton's

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta \vert m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

Can also get RMSprop or Adam

How good is this approach?

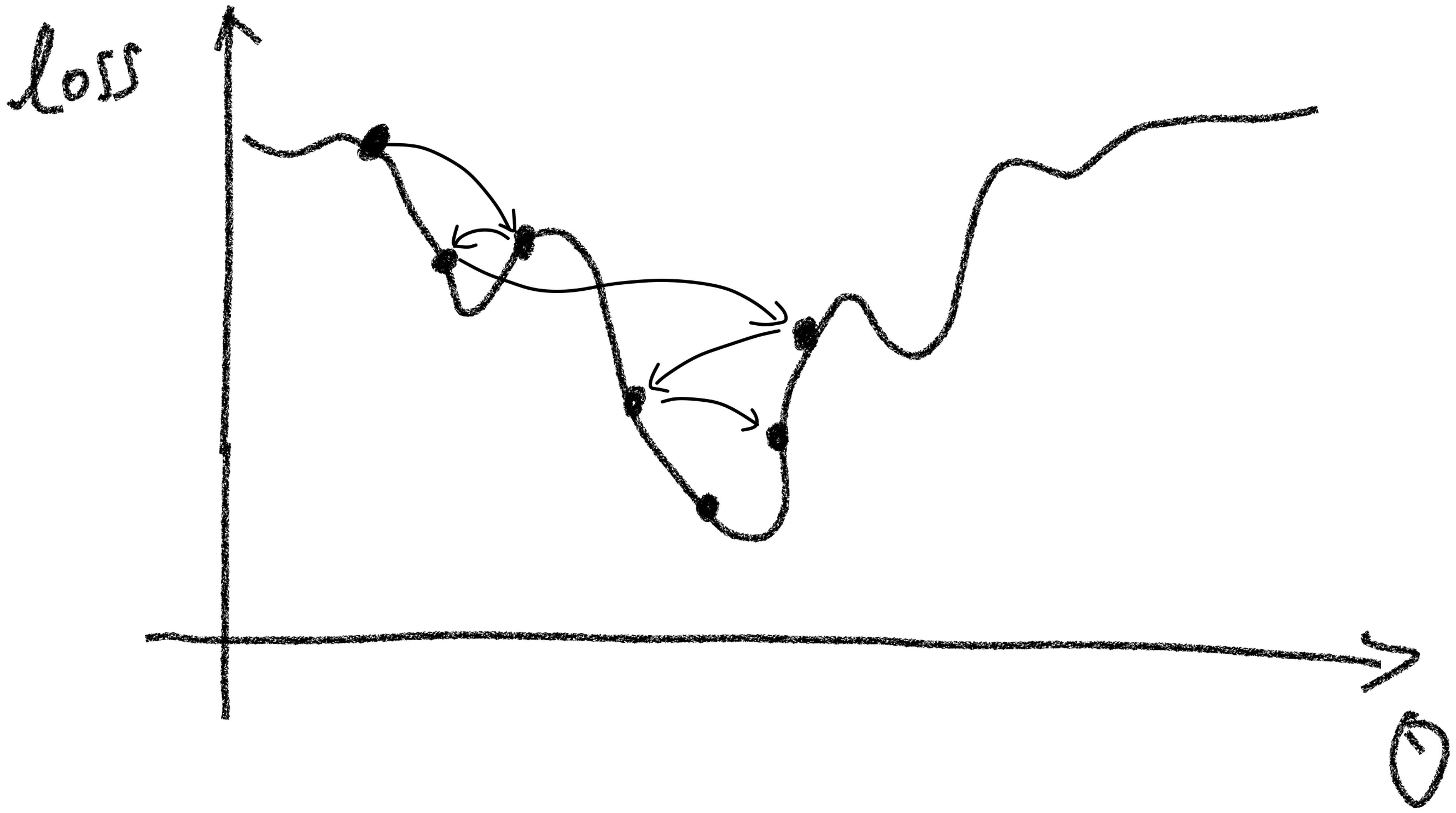
"By applying techniques such as batch normalization, data augmentation, and distributed training, we achieve similar performance in about the same number of epochs as the Adam optimizer, even on large datasets such as ImageNet."

<**Practical Deep Learning with Bayesian Principles**>, Osawa et al, NeurIPS 2019

Option 2: Interpolation-based

The idea is even simpler, when we optimize a DNN, we get a sequence of points, that is visited at different times of the optimization.

We look into the sequence, and choose some of the points to be the representative points and say this set of points approximates the posterior.



How good is this approach?

Slightly better than standard algorithm like Adam.

And compared with Variational inference (two ~ ten times computation), this is cheap.

PART 3

What goes wrong?

the theory says it should be better but wait. . .

A simple baseline

Recall that, the reason we wish to use Bayesian Deep Learning, is that we wish to have some sort of model uncertainty.

Given limited Data&Many parameters → Multiple model can fit the data well

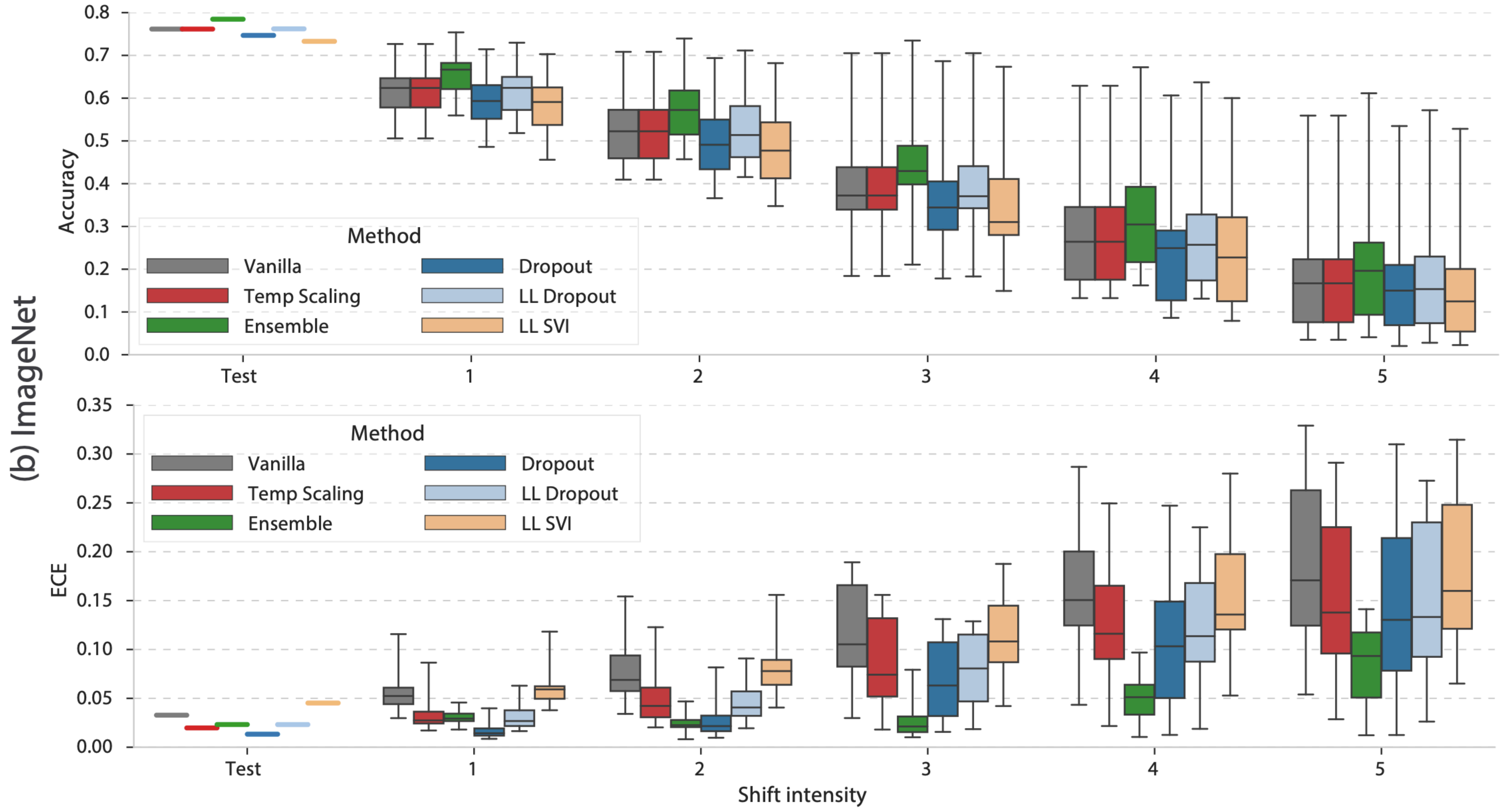
↓

Since multiple models can fit the data well, we do not just obtain one model, we instead obtain a distribution of possible models (the posterior)

A simple baseine: Deep Ensemble

Train multiple models with different initializations. And then treat this set of models as if they are sampled from the true posterior.

Deep Ensemble is good in both accuracy and uncertainty!



Left->right, shift of data

Upper figure is accuracy (higher the better)

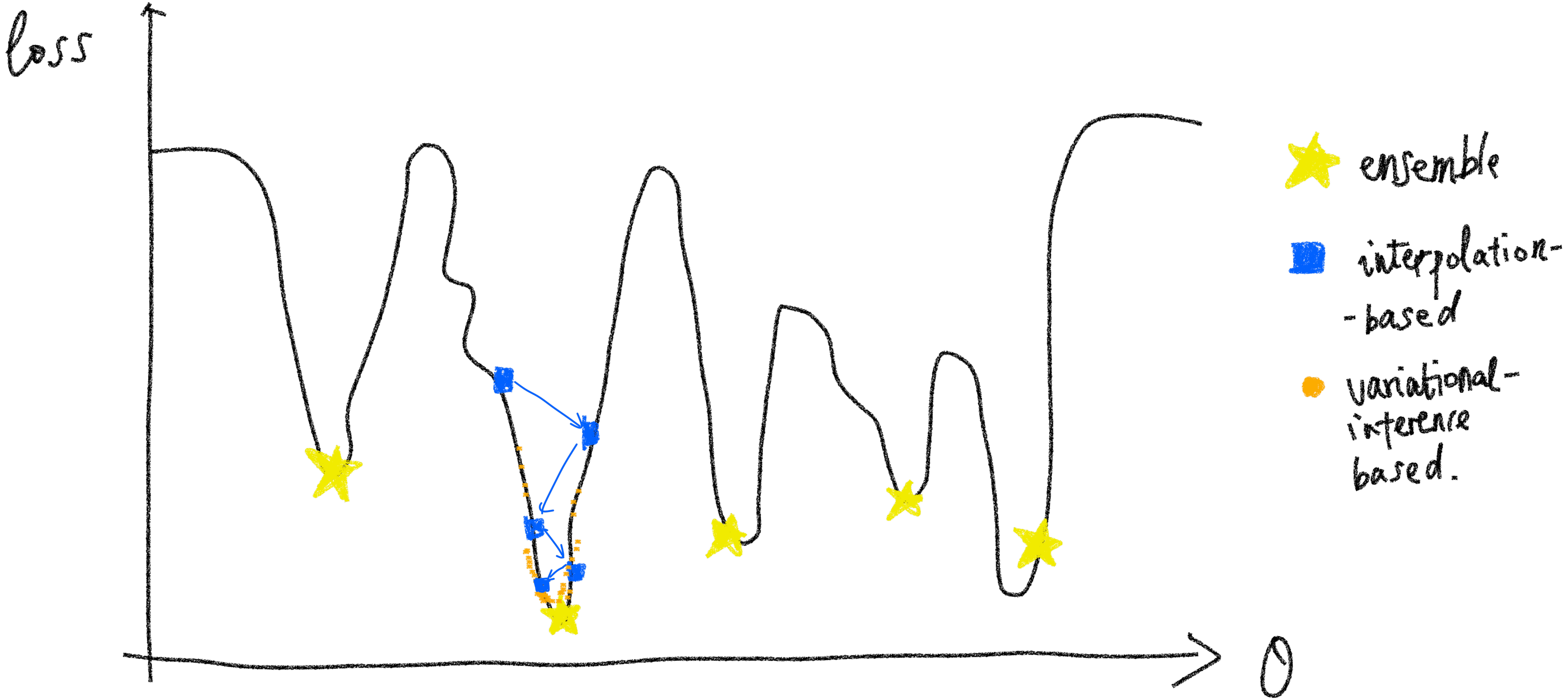
Bottom is calibration (lower the better)

SVI is variational inference based method.

Interpolation-based is not compared

Why ensemble is working?

Ensemble visits a diverse set of *Basins of attraction*



Because we know about the lottery hypothesis

The lottery hypothesis: given limited data, a over-parameterized neural network have many high-performance local minima

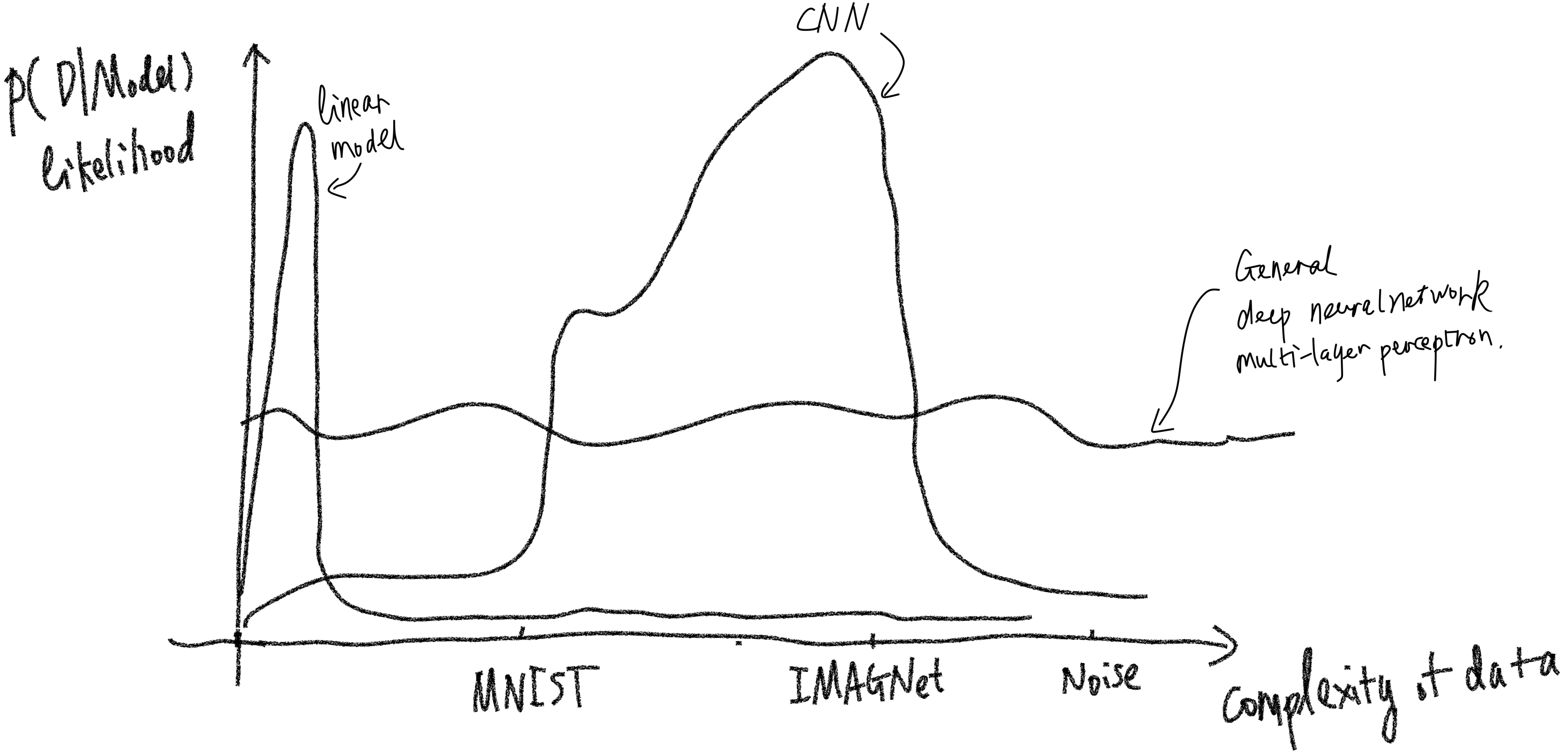
Backprop	→	Ensemble of Backprop
Variational-inference based (uni-mode **q**)	→	Variational-inference based (multi-mode **q**, such as mixture of Gaussian
Interpolation-based (single trajectory)	→	Interpolation-based (multiple trajectory)

PART 4

A New Hope

Be practical, be focused.

Prior on Structure: inductive biases



Think twice about the purpose of posterior.

Some of us are not really interested in Bayesian Neural Network, we are interest at its advantages, i.e., the posterior gives us multiple models

In general, If resource permitted, just use ensemble!

Transfer or meta-learning

Transfer or meta-learning is exactly where we need a posterior distribution, instead of just a single model.

Summary

PART 1: Combining Bayesian and Deep Learning

motivation: obtain a posterior distribution of models, not just a single model.

PART 2: Bayesian treatment of Deep Learning

Two technical approach: variational-inference based, and interpolation-based

PART 3: What goes wrong?

Why Part 2 is not working great compared with a simple baseline: ensemble

PART 4: A New Hope

Advice and opinion (personal idea)

1. Also consider the possible architectures, not just the weights

2. Consider the potential benefits of posterior, not just in standard setting, but for transfer or meta learning.

Thank you.