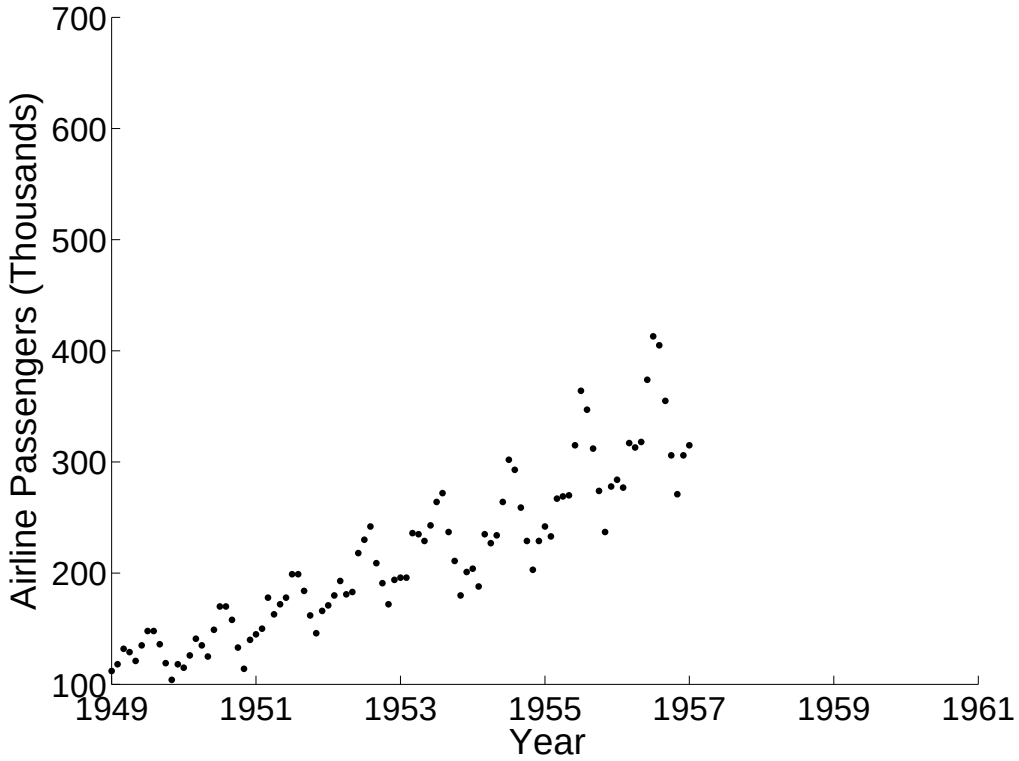
Loss Valleys and Generalization in Deep Learning

Andrew Gordon Wilson

Assistant Professor
https://people.orie.cornell.edu/andrew
Cornell University

The Robotic Vision Probabilistic Object Detection Challenge
CVPR
Long Beach, CA
June 17, 2019

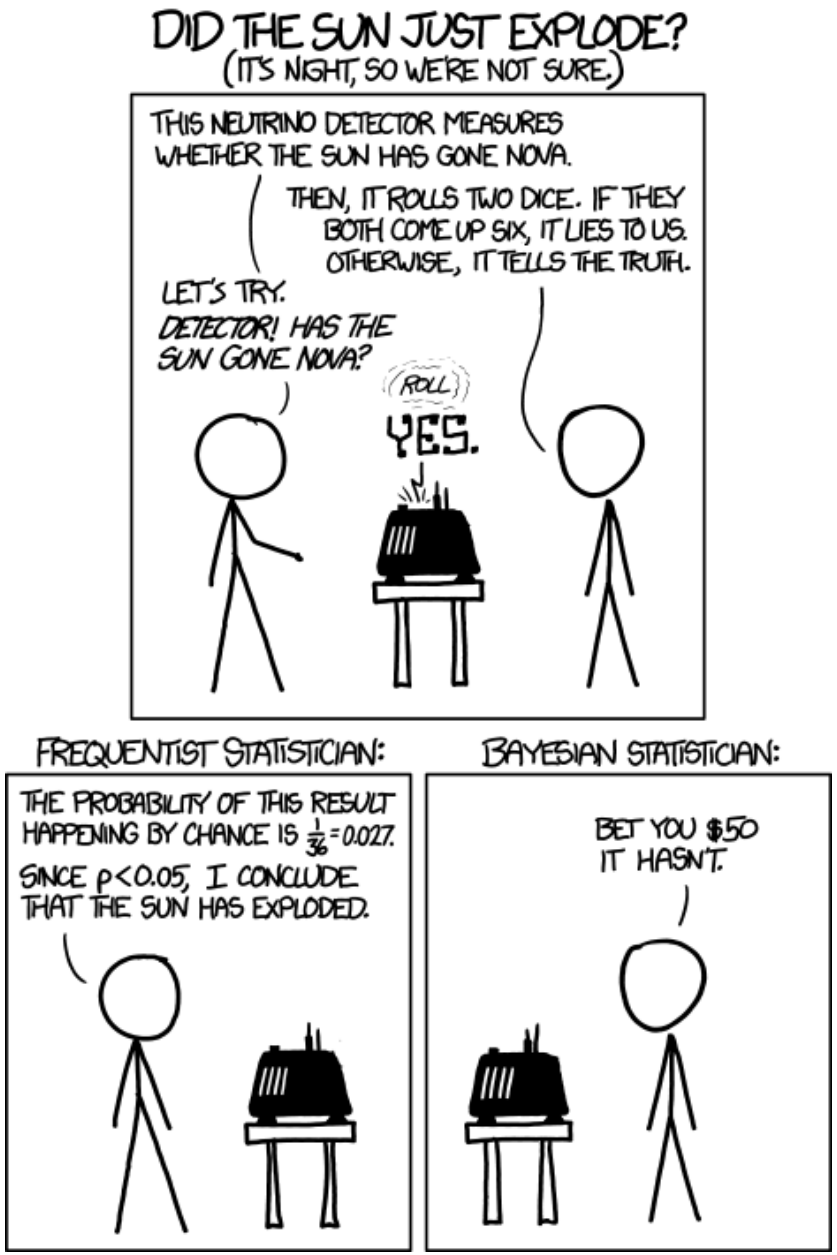
Model Selection



Which model should we choose?

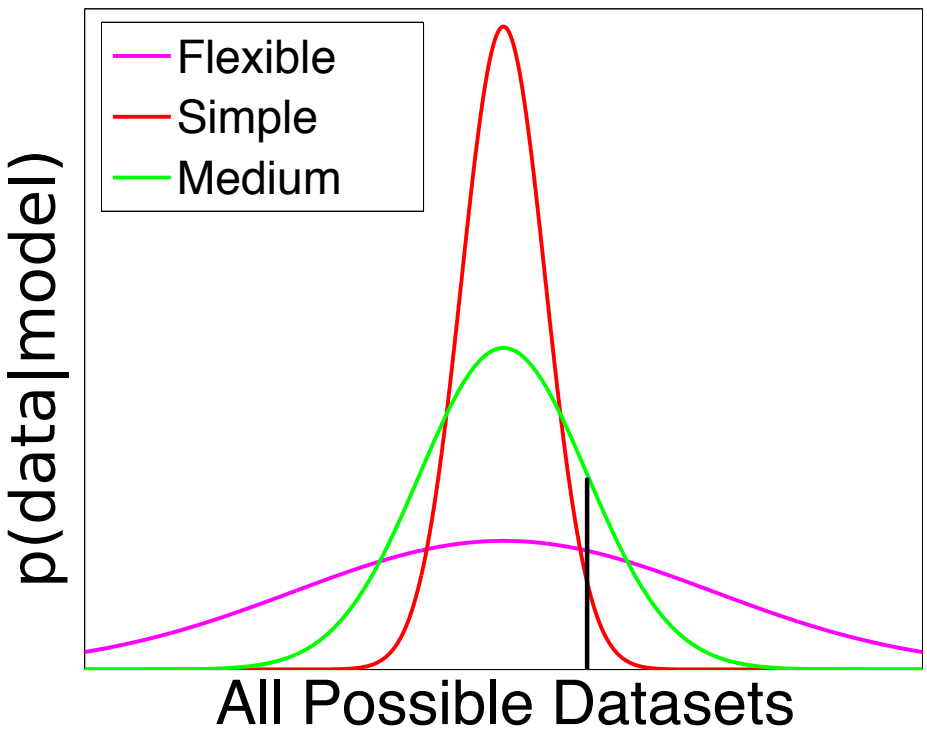
(1): $f_1(x) = a_0 + a_1 x$ **(2):** $f_2(x) = \sum_{j=0}^{3} a_j x^j$ **(3):** $f_3(x) = \sum_{j=0}^{10^4} a_j x^j$

Bayesian or Frequentist?



How do we learn?

- The ability for a system to learn is determined by its *support* (which solutions are a priori possible) and *inductive biases* (which solutions are a priori likely).
- An influx of new *massive* datasets provide great opportunities to automatically learn rich statistical structure, leading to new scientific discoveries.



Bayesian Deep Learning

Why?

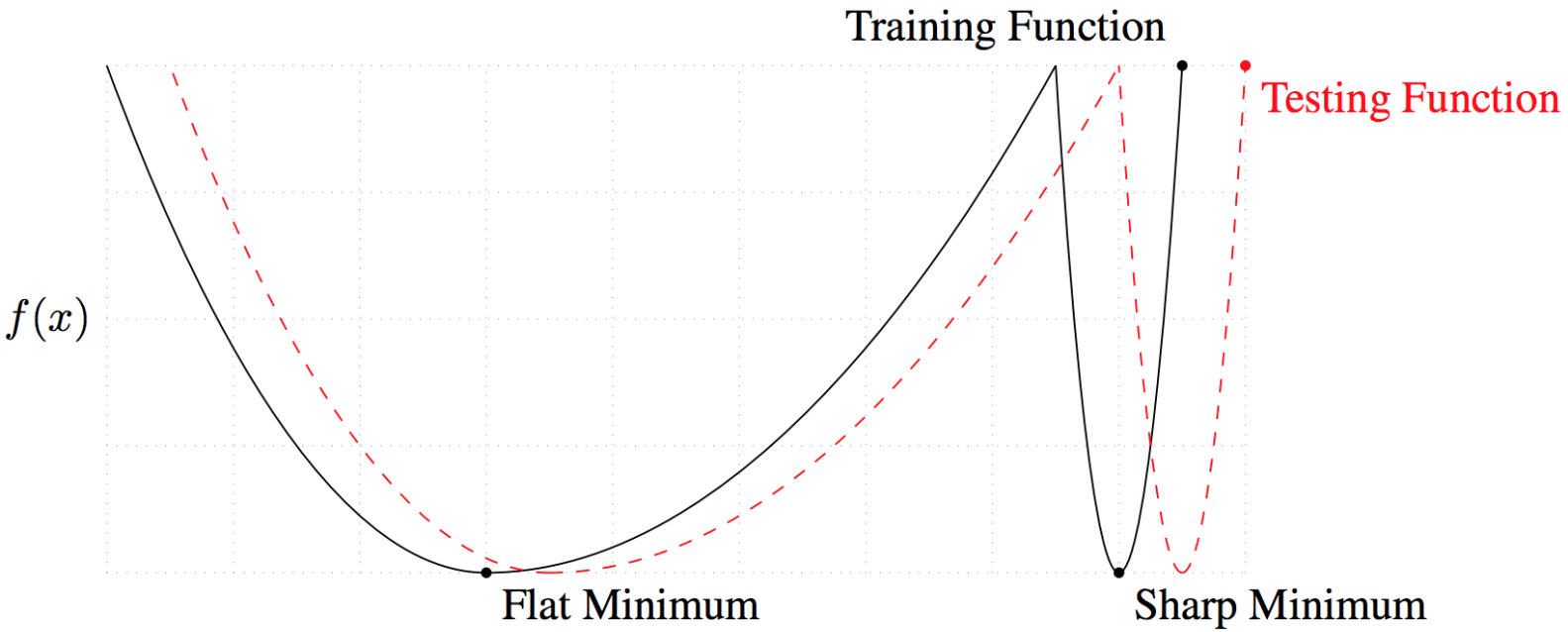
- A powerful framework for model construction and understanding generalization
- Uncertainty representation and calibration (crucial for decision making)
- *Better point estimates*
- Interpretably incorporate prior knowledge and domain expertise
- It was the most successful approach at the end of the second wave of neural networks (Neal, 1998).
- Neural nets are much less mysterious when viewed through the lens of probability theory.

Why not?

- Can be computationally intractable (but doesn't have to be).
- Can involve a lot of moving parts (but doesn't have to).

There has been exciting progress in the last two years addressing these limitations as part of an extremely fruitful research direction.

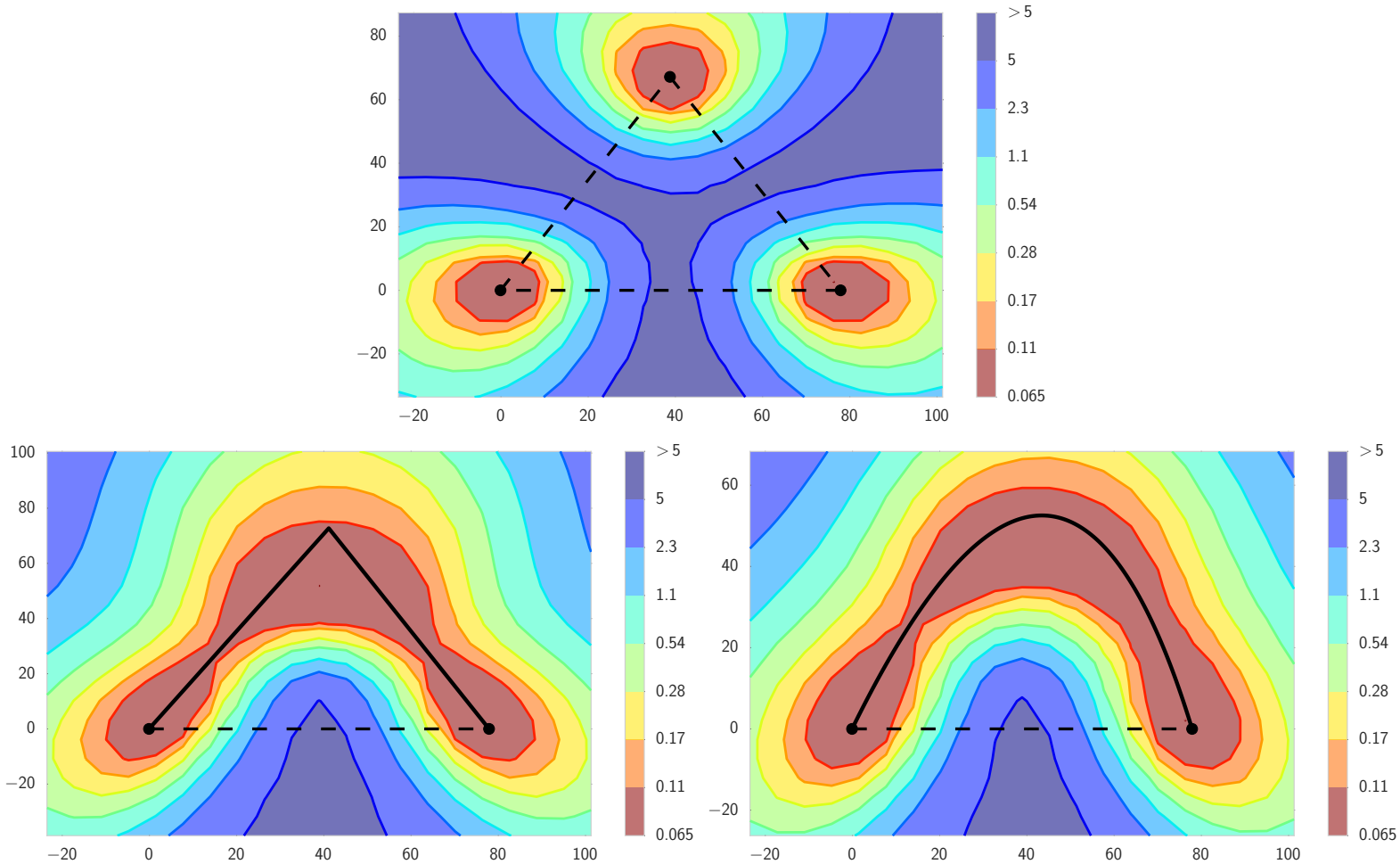
Wide Optima Generalize Better



Keskar et. al (2017)

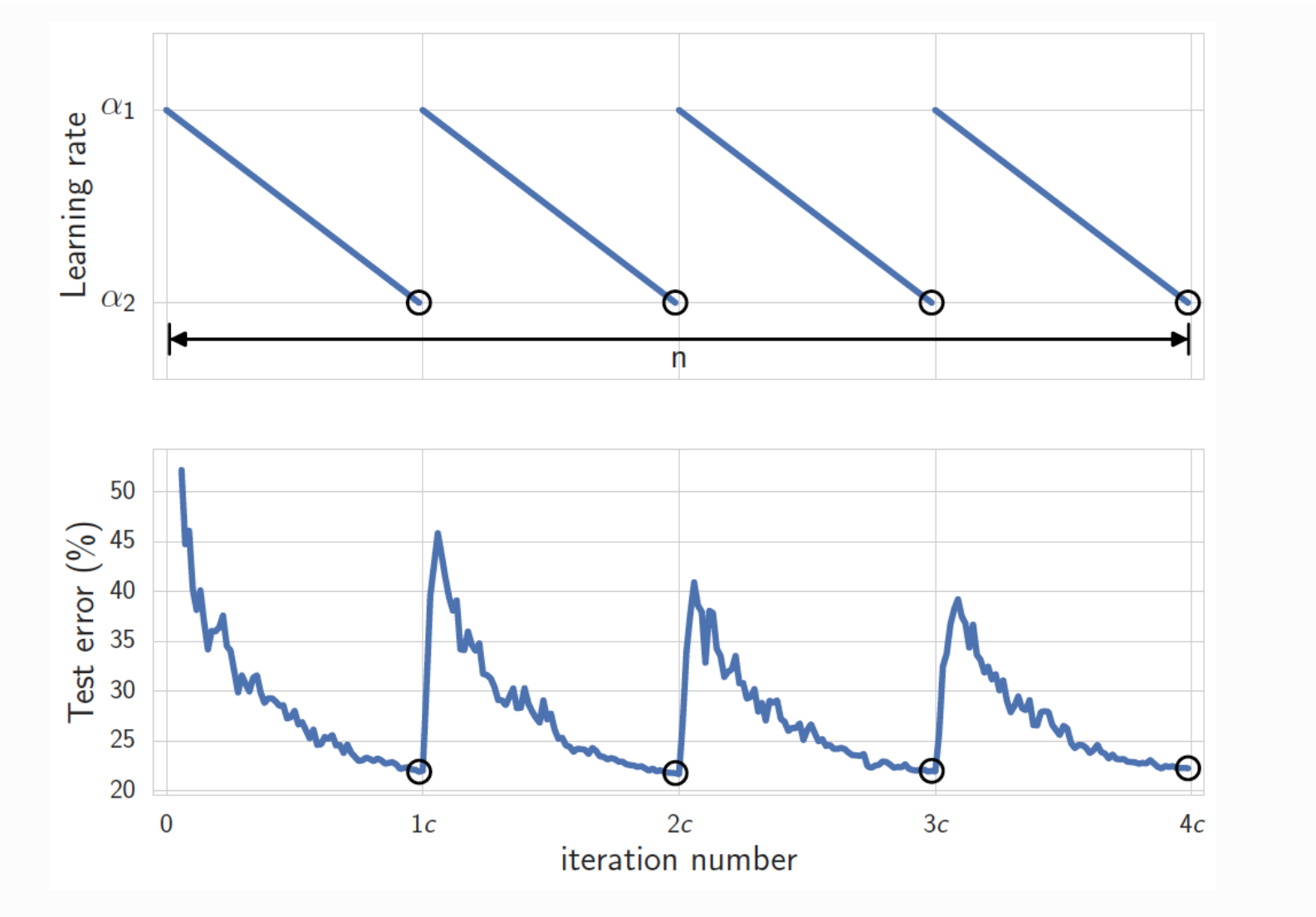
- ***Bayesian integration* will give *very* different predictions *in deep learning especially*!**

Mode Connectivity

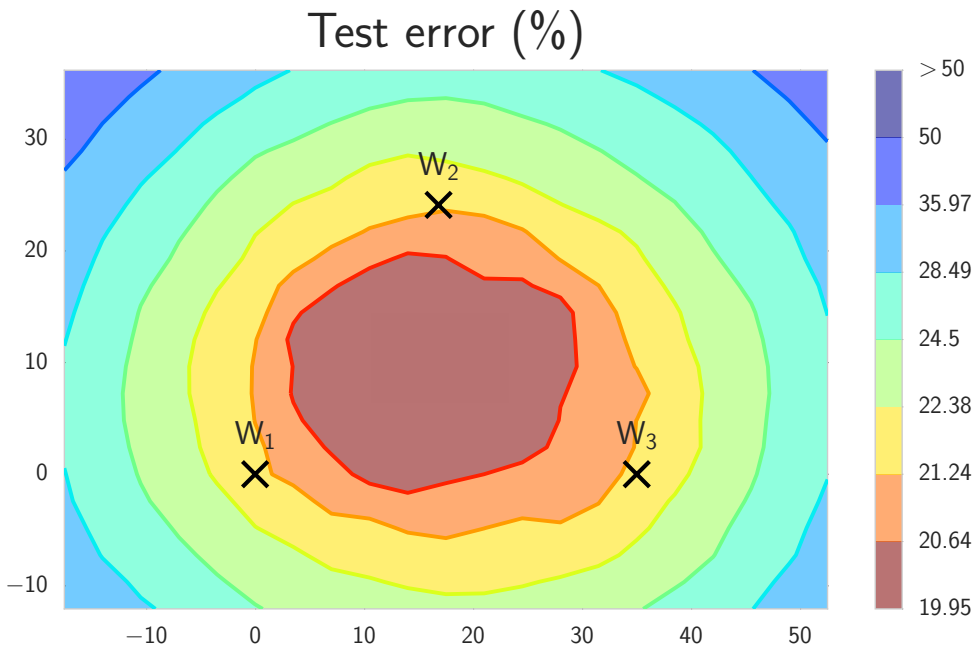


Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs
Advances in Neural Information Processing Systems (NeurIPS), 2018
T. Garipov, P. Izmailov, D. Podoprikhin, D. Vetrov, A.G. Wilson

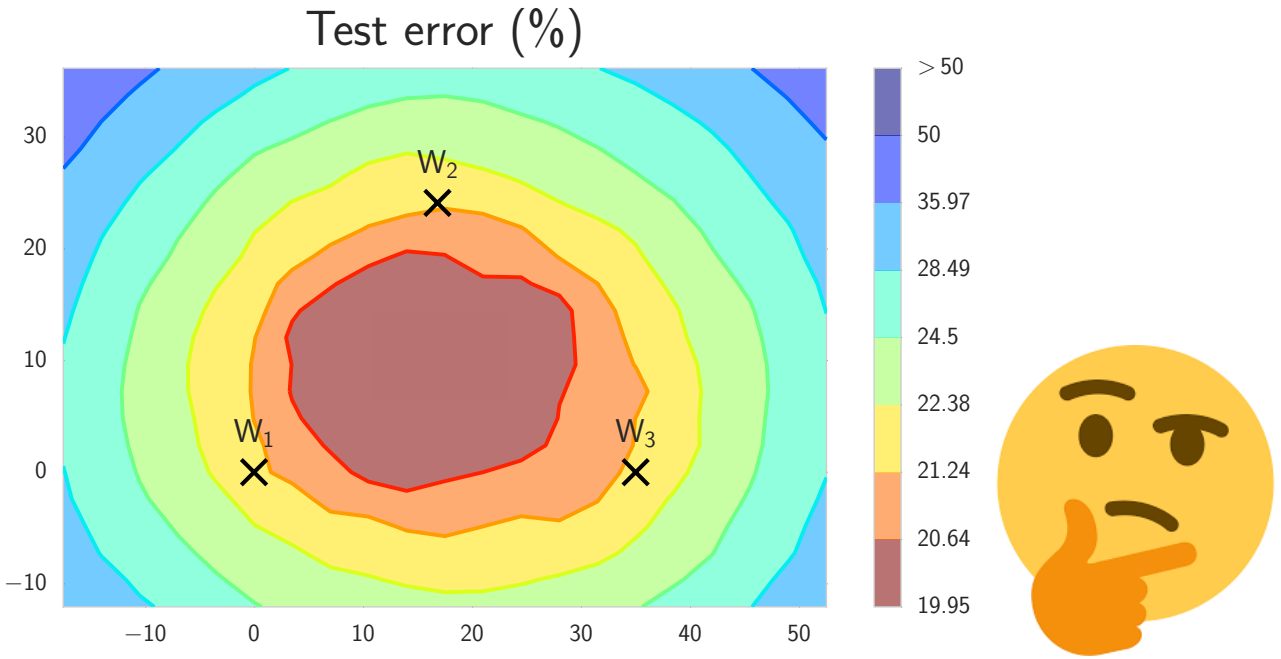
Cyclical Learning Rate Schedule



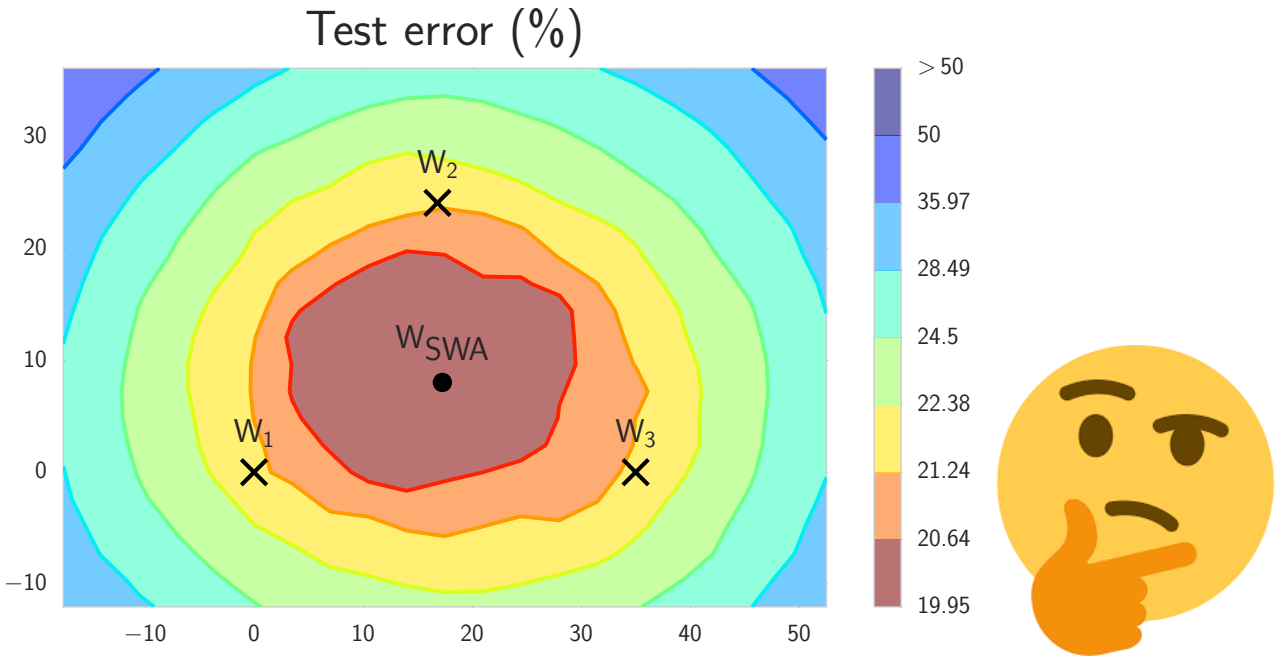
Trajectory of SGD



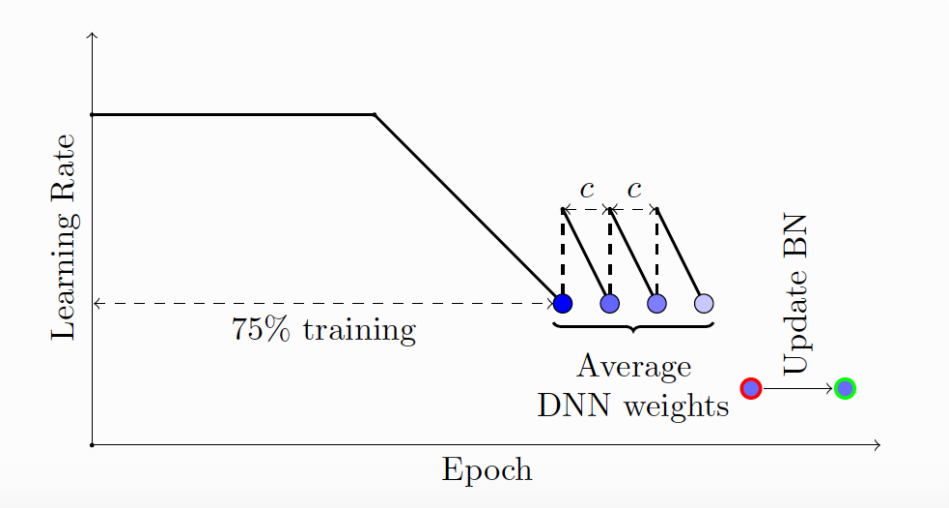
Trajectory of SGD

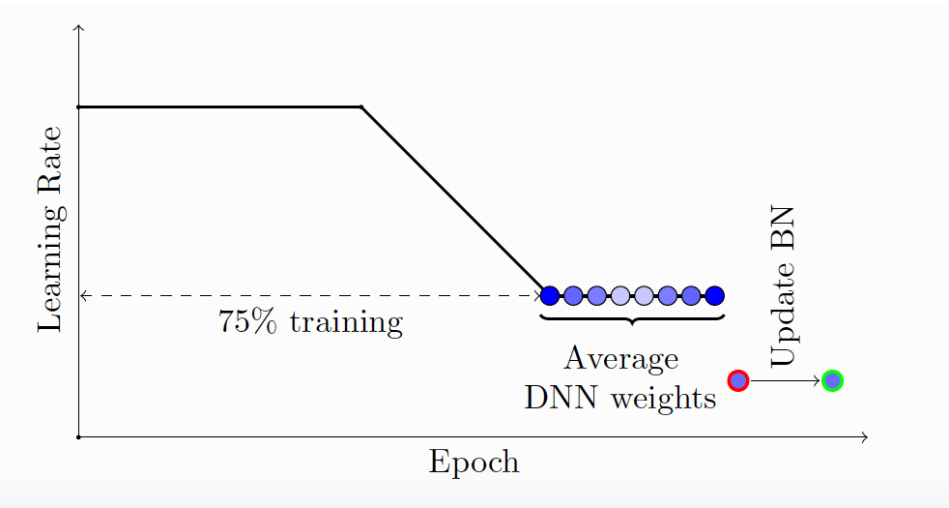


Trajectory of SGD



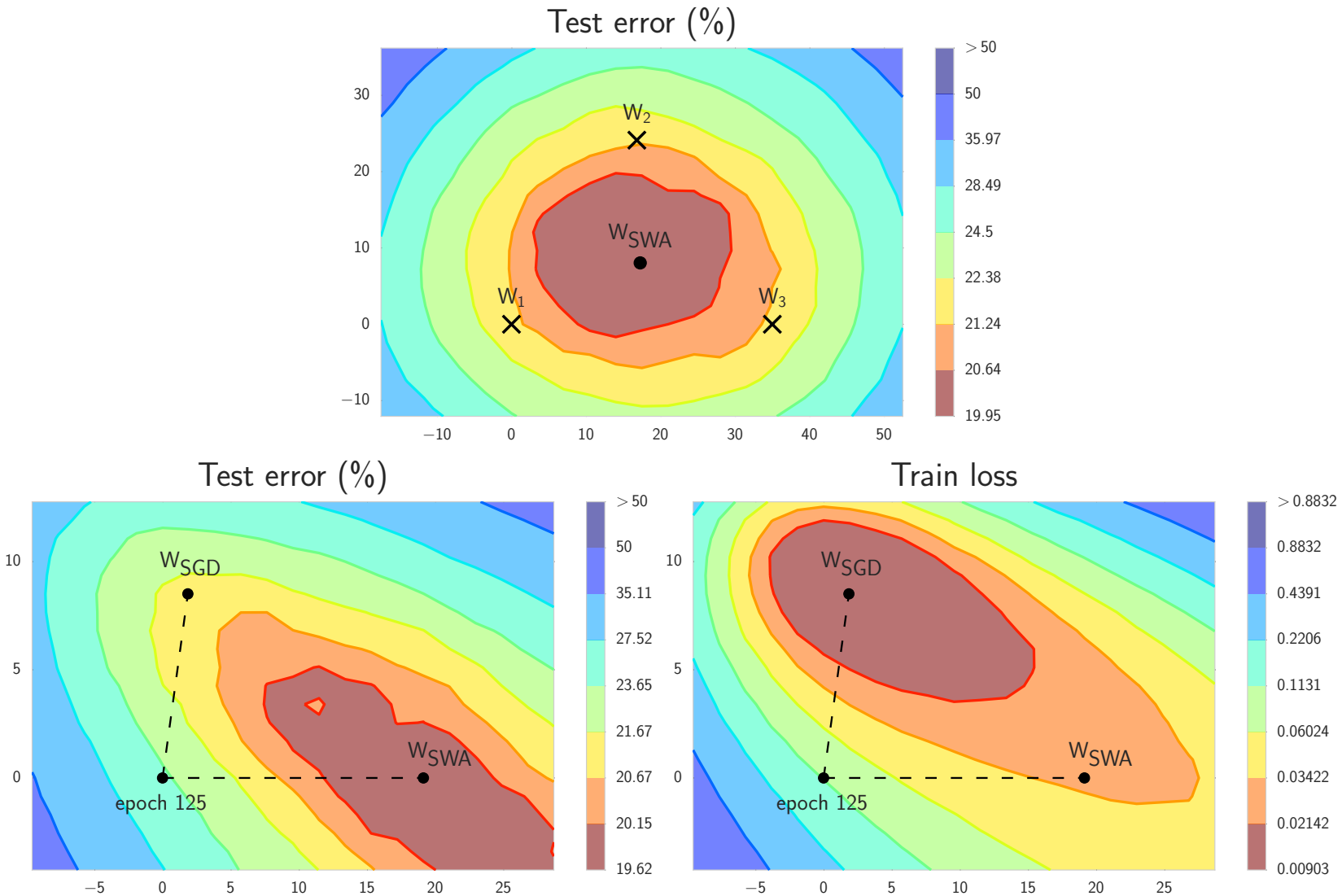
SWA Algorithm



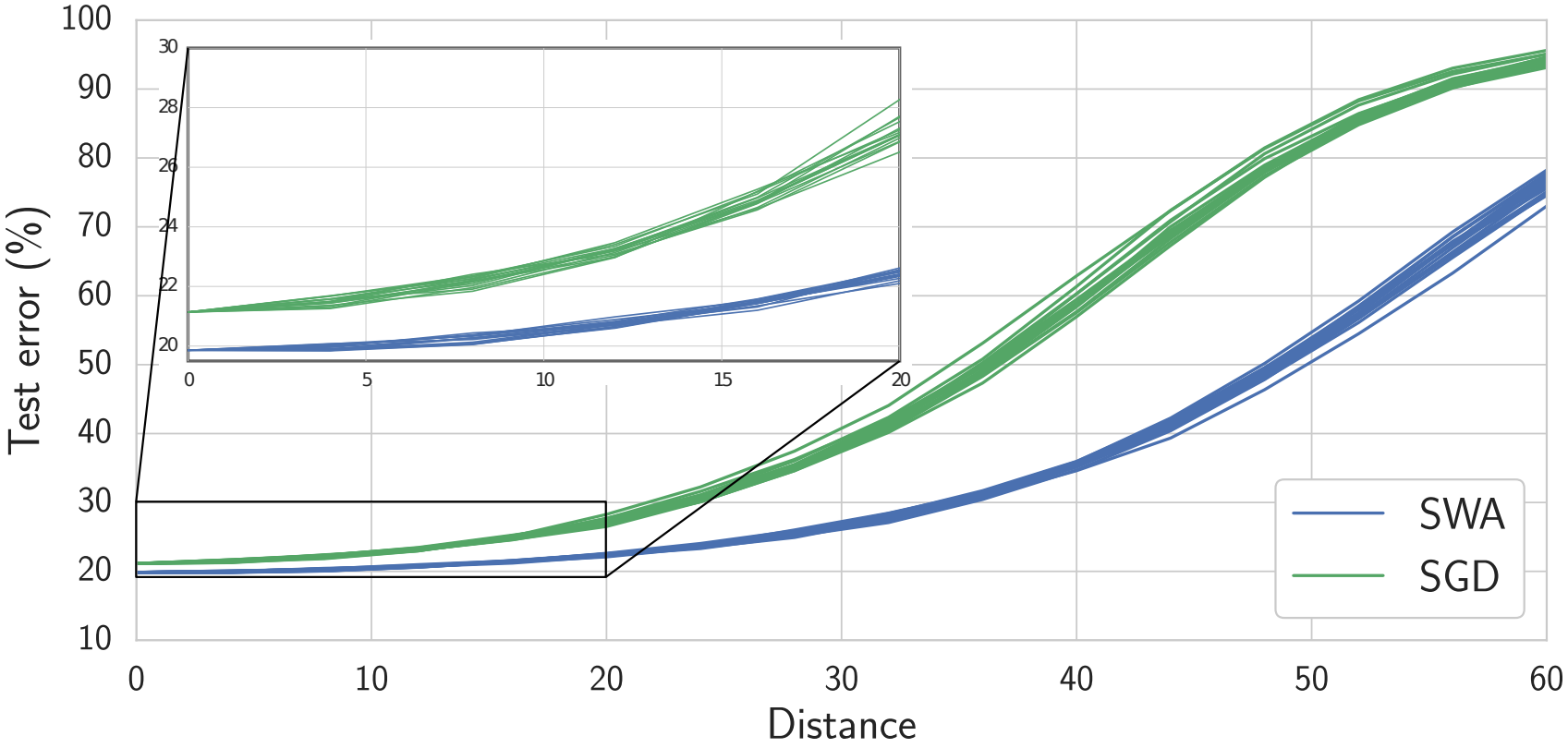


- Use learning rate that doesn't decay to zero (cyclical or constant)
- Average weights
  - Cyclical LR: at the end of each cycle
  - Constant LR: at the end of each epoch
- Recompute batch normalization statistics at the end of training; in practice, do one additional forward pass on the training data.

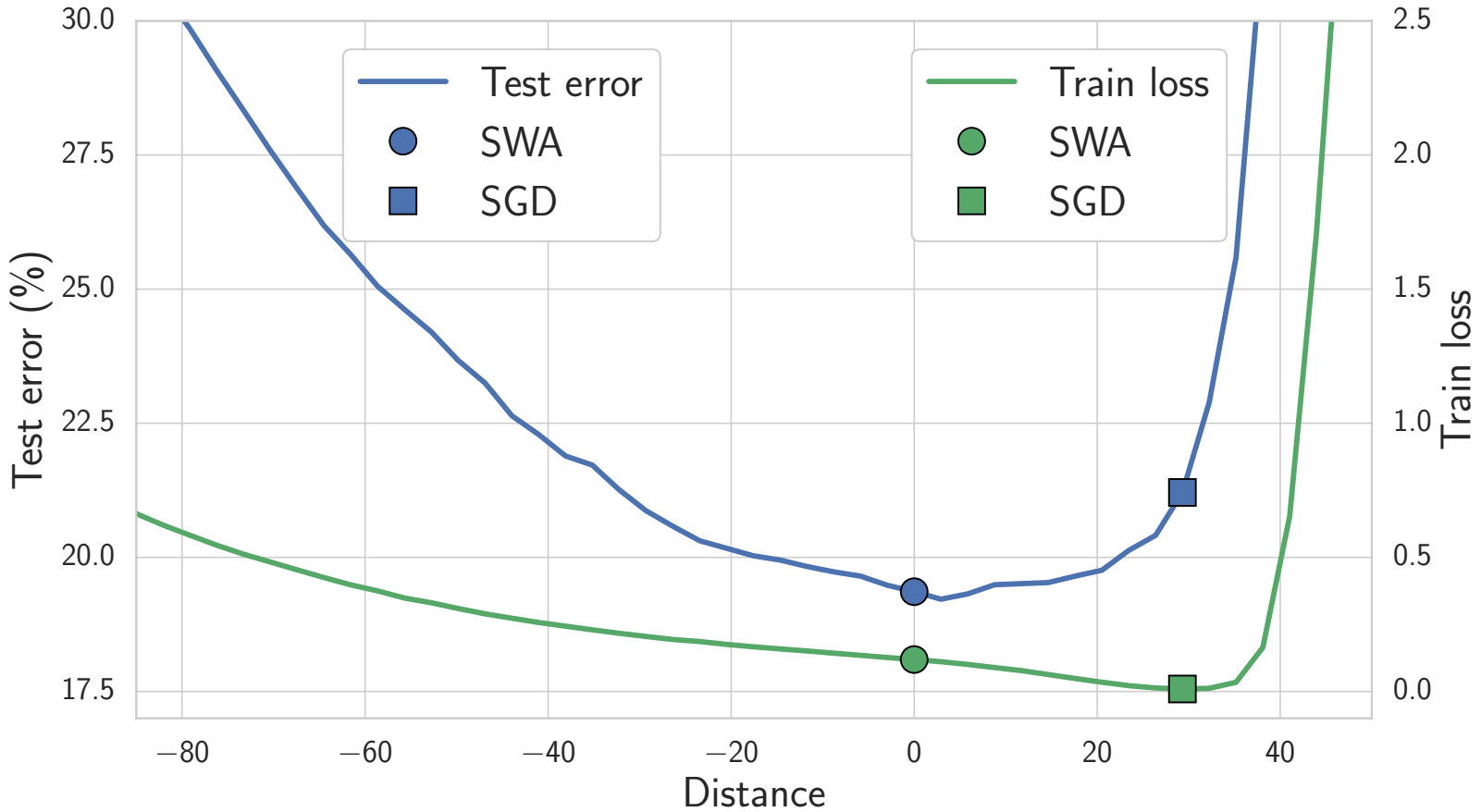
Trajectory of SGD



Following Random Paths



Path from w_{SWA} to w_{SGD}



Approximating an FGE Ensemble

Because the points sampled from an FGE ensemble take small steps in weight space *by design*, we can do a linearization analysis to show that

$$f(w_{\text{SWA}}) \approx \frac{1}{n} \sum f(w_i)$$

SWA Results, CIFAR

Table 1: Accuracies (%) of SWA, SGD and FGE methods on CIFAR-100 and CIFAR-10 datasets for different training budgets. Accuracies for FGE were taken from [Garipov et al., 2018].

			SWA		
DNN (Budget)	SGD	FGE (1 Budget)	1 Budget	1.25 Budgets	1.5 Budgets
		CIFAR-100			
VGG-16 (200)	72.55 ± 0.10	74.26	73.91 ± 0.12	74.17 ± 0.15	74.27 ± 0.25
ResNet-110 (150)	78.49 ± 0.36	79.84	79.77 ± 0.17	80.18 ± 0.23	80.35 ± 0.16
WRN-28-10 (200)	80.82 ± 0.23	82.27	81.46 ± 0.23	81.91 ± 0.27	82.15 ± 0.27
PyramidNet-272 (300)	83.41 ± 0.21	–	–	83.93 ± 0.18	84.16 ± 0.15
		CIFAR-10			
VGG-16 (200)	93.25 ± 0.16	93.52	93.59 ± 0.16	93.70 ± 0.22	93.64 ± 0.18
ResNet-110 (150)	95.28 ± 0.10	95.45	95.56 ± 0.11	95.77 ± 0.04	95.83 ± 0.03
WRN-28-10 (200)	96.18 ± 0.11	96.36	96.45 ± 0.11	96.64 ± 0.08	96.79 ± 0.05
ShakeShake-2x64d (1800)	96.93 ± 0.10	–	–	97.16 ± 0.10	97.12 ± 0.06

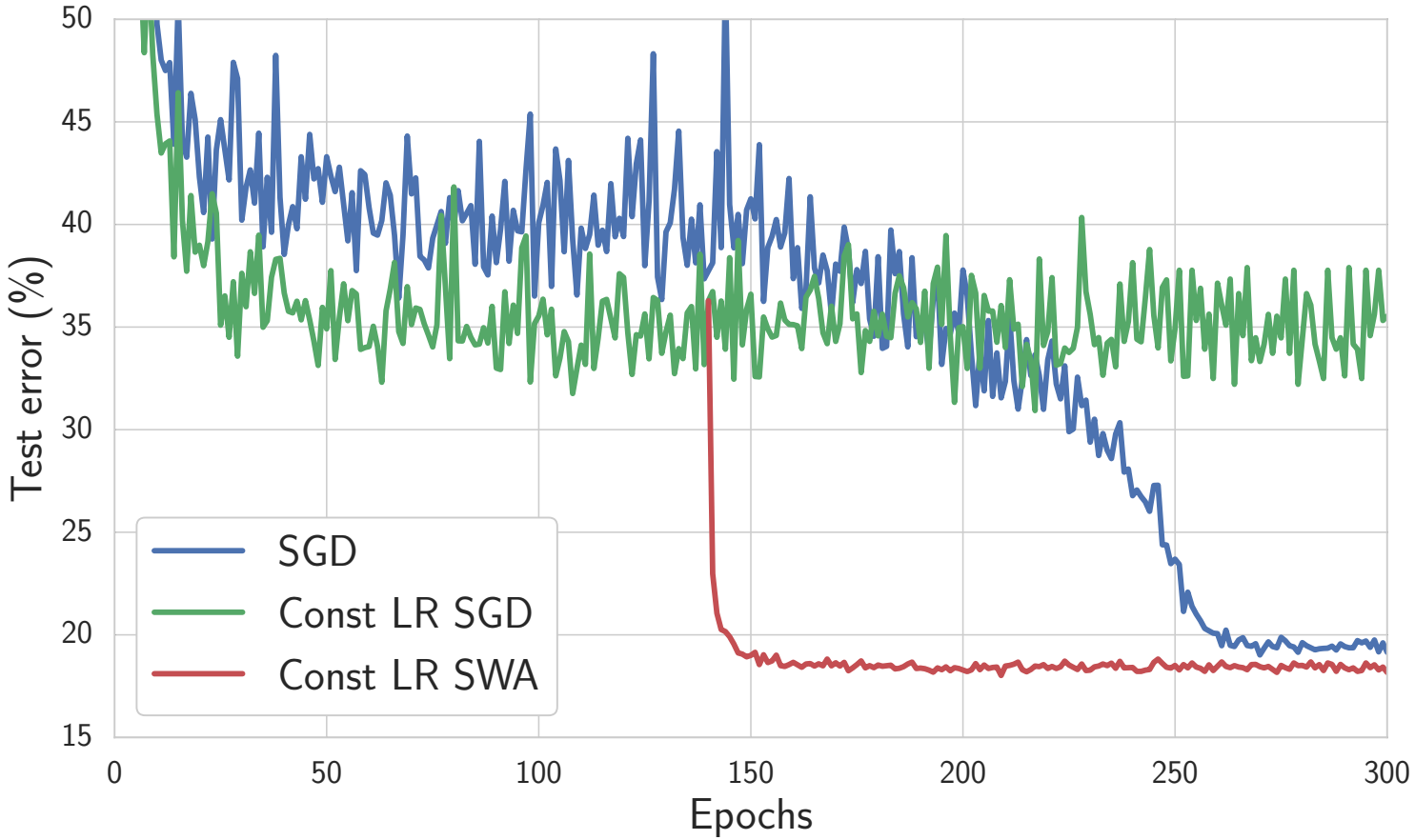
SWA Results, ImageNet (Top-1 Error Rate)

		SWA	
DNN	SGD	5 epochs	10 epochs
ResNet-50	76.15	76.83 ± 0.01	76.97 ± 0.05
ResNet-152	78.31	78.82 ± 0.01	78.94 ± 0.07
DenseNet-161	77.65	78.26 ± 0.09	78.44 ± 0.06

Sampling from a High Dimensional Gaussian

SGD (with constant LR) proposals are on the surface of a hypersphere. Averaging lets us go inside the sphere to a point of higher density.

High Constant LR



Side observation: Averaging bad models does not give good solutions. Averaging bad weights can give great solutions.

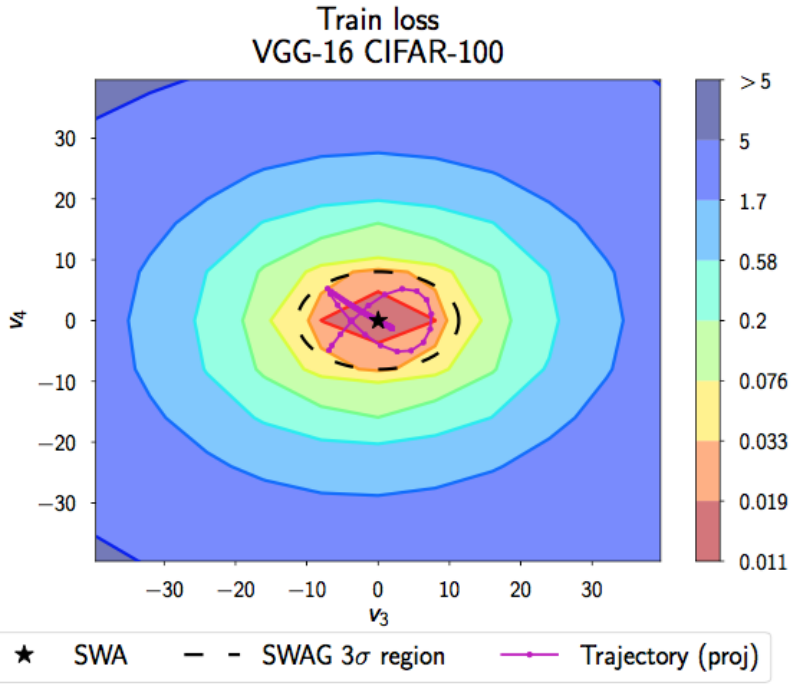
Stochastic Weight Averaging

- Simple drop-in replacement for SGD or other optimizers
- Works by finding flat regions of the loss surface
- No runtime overhead, but often significant improvements in generalization for many tasks
- Available in PyTorch contrib (call `optim.swa`)
- `https://people.orie.cornell.edu/andrew/code`

Averaging Weights Leads to Wider Optima and Better Generalization, UAI 2018
P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, A.G. Wilson.

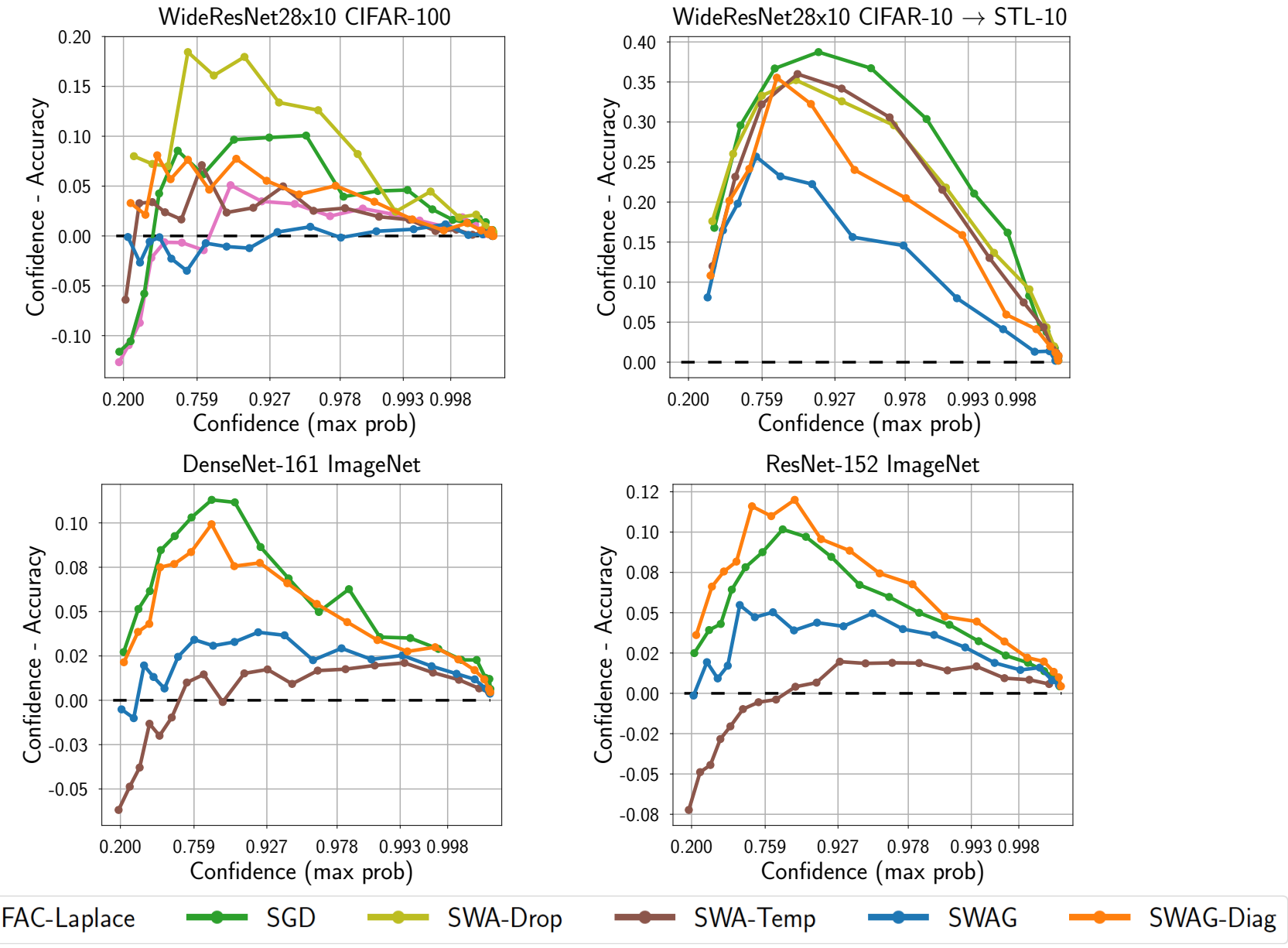
Uncertainty Representation with SWAG

1. Leverage theory that shows SGD with a constant learning rate is approximately sampling from a Gaussian distribution.
2. Compute first *two* moments of SGD trajectory (SWA computes just the first).
3. Use these moments to construct a Gaussian approximation in weight space.
4. Sample from this Gaussian distribution, pass samples through predictive distribution, and form a Bayesian model average.

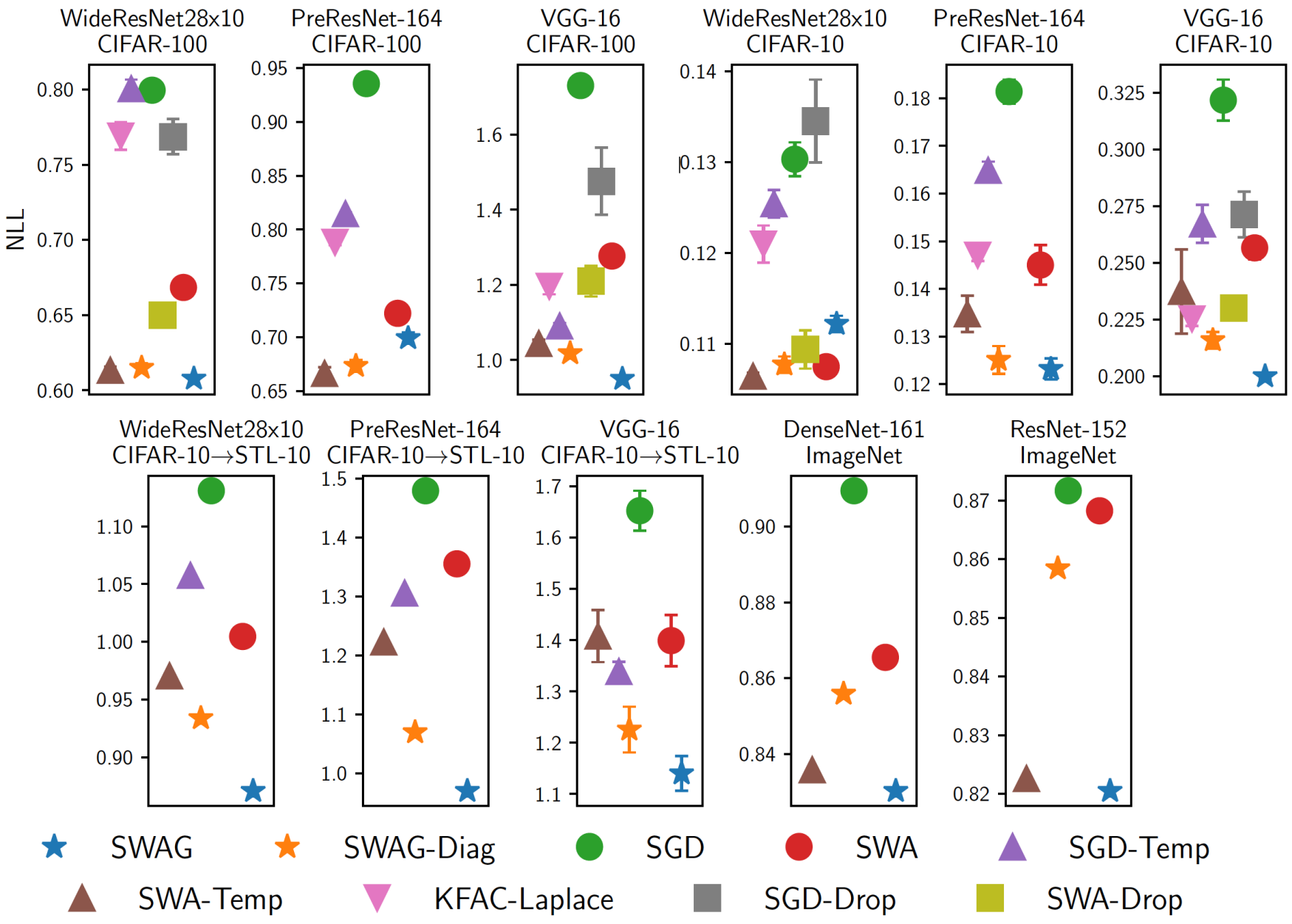


A Simple Baseline for Bayesian Uncertainty in Deep Learning
W. Maddox, P. Izmailov, T. Garipov, D. Vetrov, A.G. Wilson

Uncertainty Calibration



Uncertainty Likelihood



Subspace Inference for Bayesian Deep Learning

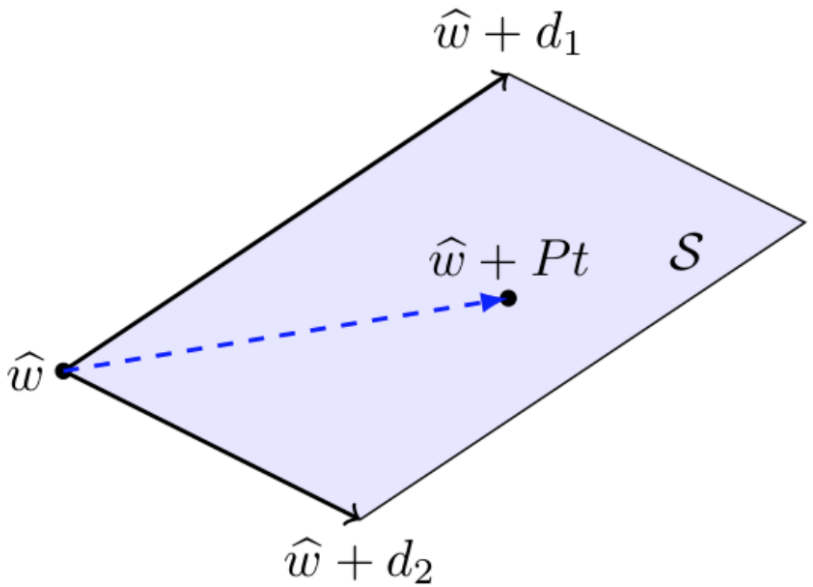
A modular approach:

- Construct a subspace of a network with a high dimensional parameter space
- Perform inference directly in the subspace
- Sample from approximate posterior for Bayesian model averaging

We can approximate the posterior of a WideResNet with 36 million parameters in a 5D subspace and achieve state-of-the-art results!

Subspace Construction

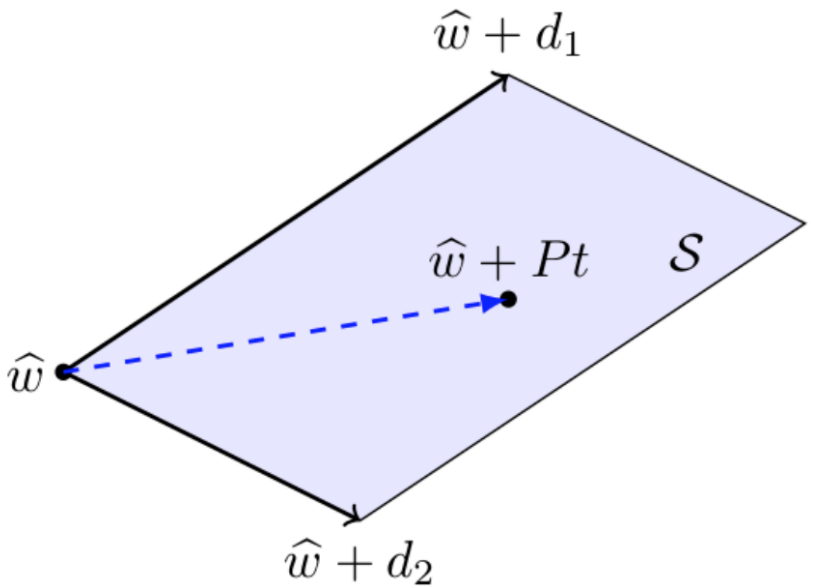
- Choose shift $\hat{w}$ and basis vectors $\{d_1, \ldots, d_k\}$.
- Define subspace $S = \{w | w = \hat{w} + t_1 d_1 + t_k d_k\}$.
- Likelihood $p(\mathcal{D}|t) = p_M(\mathcal{D}|w = \hat{w} + Pt)$.



Inference

- Approximate inference over parameters t
  - MCMC, Variational Inference, Normalizing Flows, ...
- Bayesian model averaging at test time:

$$p(\mathcal{D}_*|\mathcal{D}) = \frac{1}{J}\sum_{j=1}^{J} p_M(\mathcal{D}_*|\tilde{w} = \hat{w} + P\tilde{t}_i)\,, \quad \tilde{t}_i \sim q(t|\mathcal{D}) \tag{1}$$



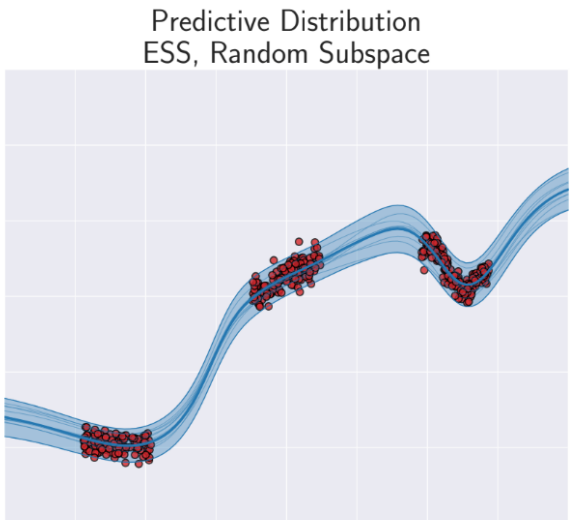
Subspace Choice

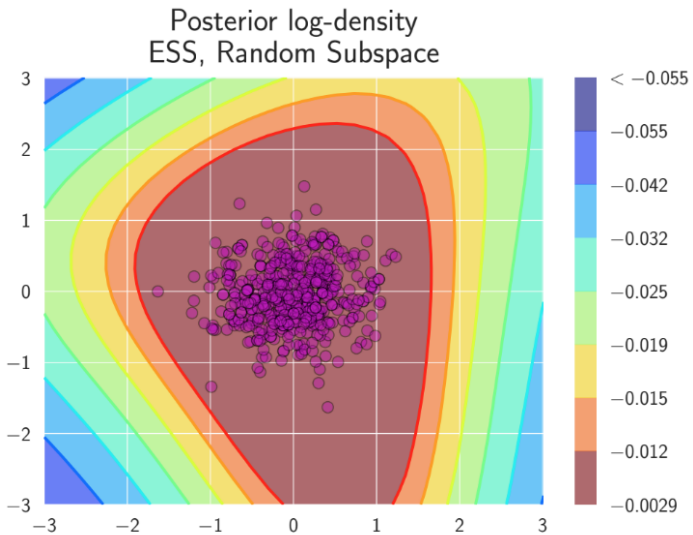
We want a subspace that

- Contains **diverse** models which give rise to different predictions
- **Cheap** to construct

Random Subspace

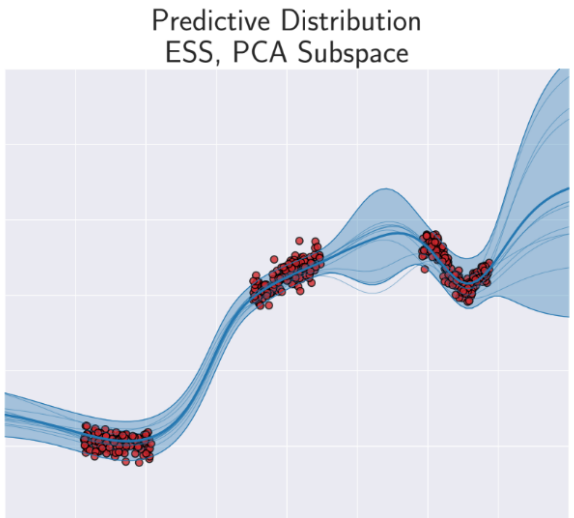
- Directions $d_1, \ldots, d_k \sim \mathcal{N}(0, I_p)$
- Use pre-trained solution as shift $\hat{w}$
- Subspace $S = \{w | w = \hat{w} + Pt\}$

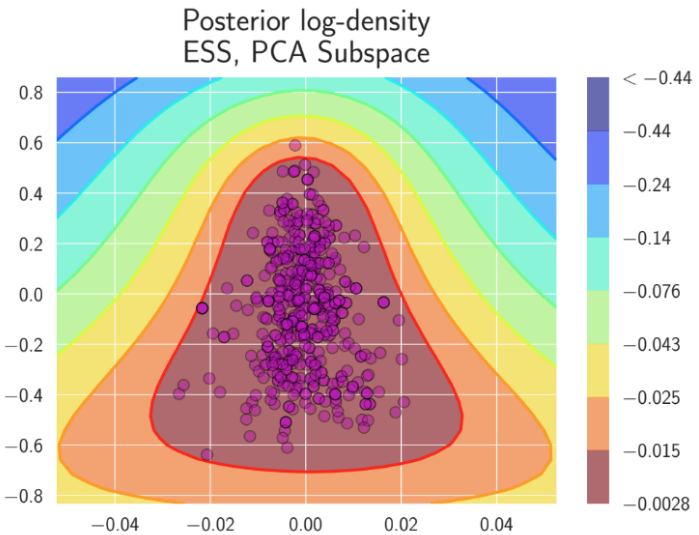




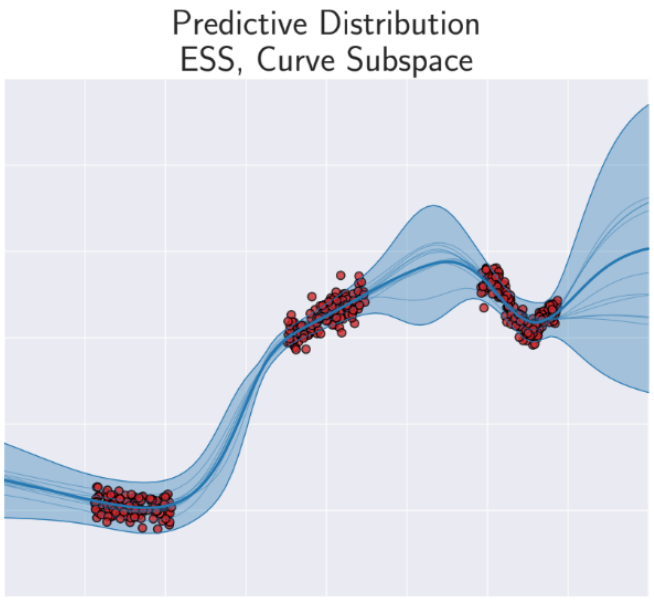
PCA of SGD Trajectory

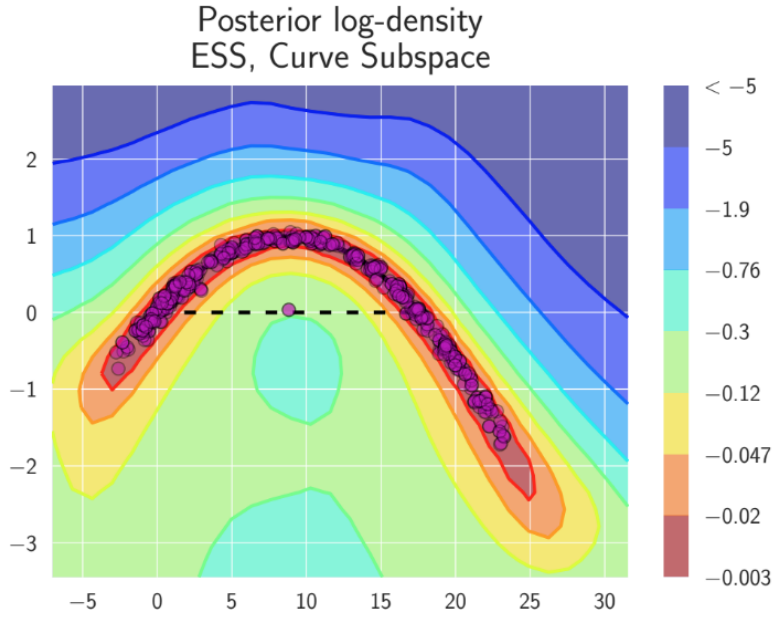
- Run SGD with a high constant learning rate from a pre-trained solution
- Collect snapshots of weights w_i
- Use SWA solution as shift $\hat{w} = \frac{1}{M} \sum_i w_i$
- $\{d_1, \ldots, d_k\}$ are the first k PCA components of vectors $\hat{w} - w_i$.



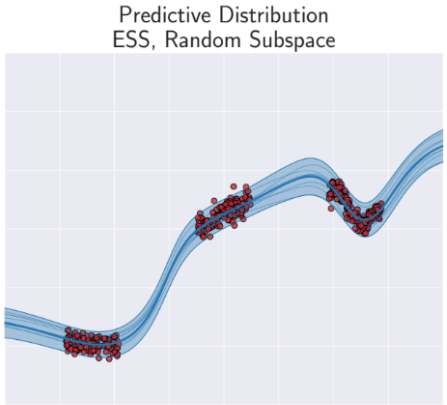


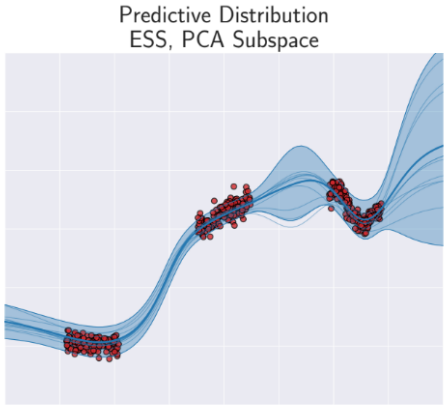
Curve Subspace

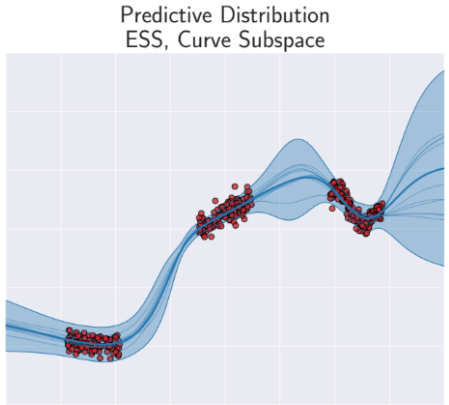


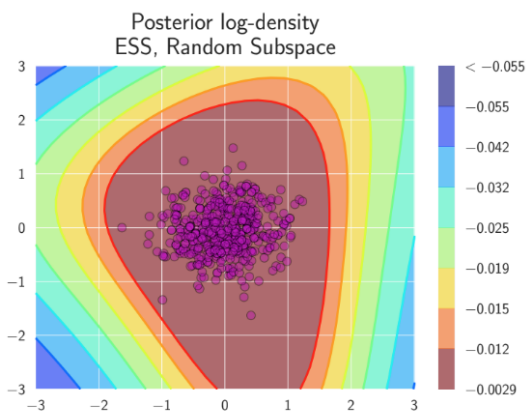


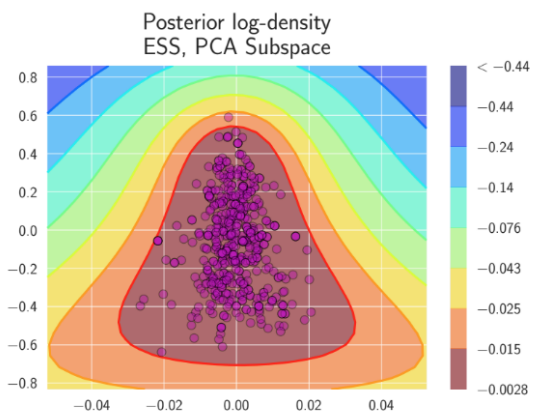
Subspace Comparison (Regression)

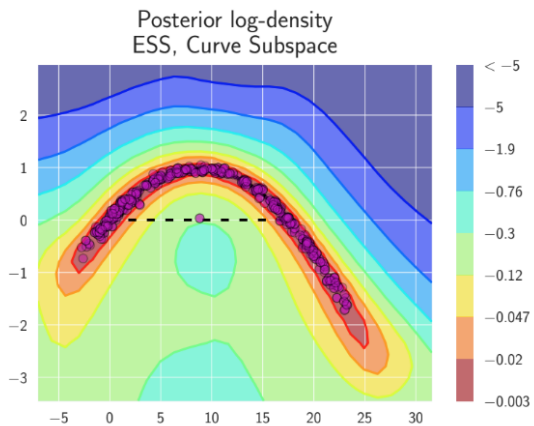




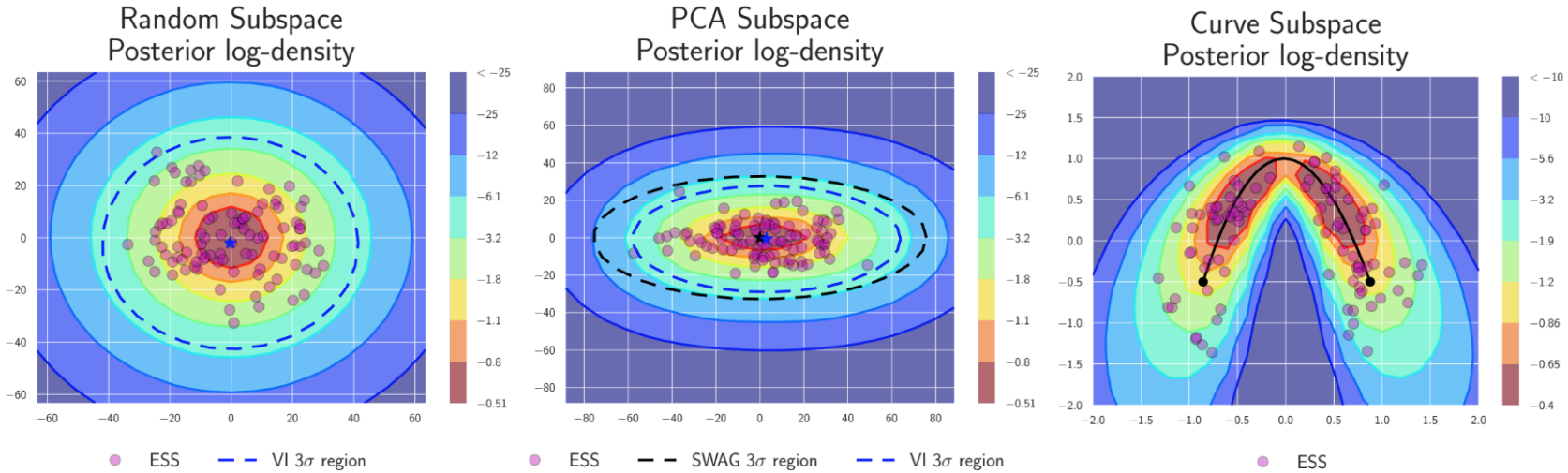








Subspace Comparison (Classification)



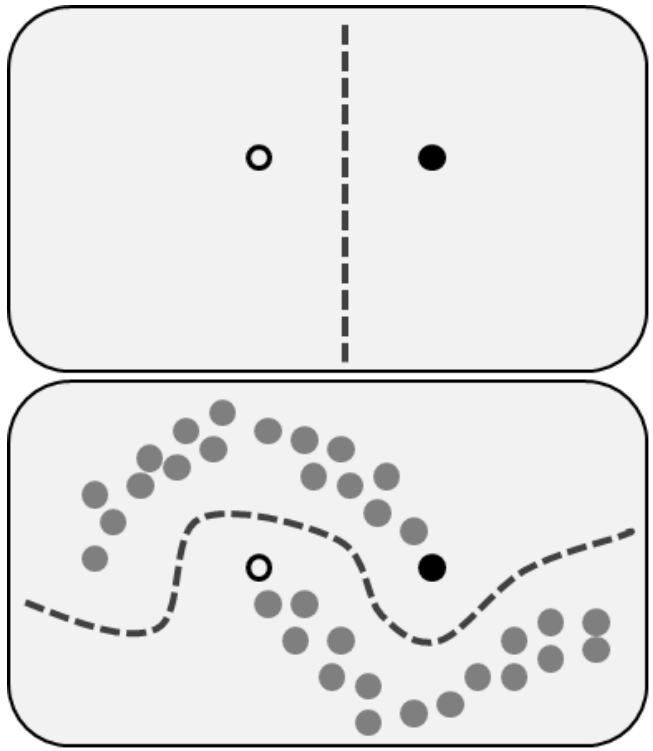
	SGD	Random	PCA	Curve
NLL	0.946 ± 0.001	0.686 ± 0.005	0.665 ± 0.004	0.646
Accuracy (%)	78.50 ± 0.32	80.17 ±0.03	80.54 ± 0.13	81.28

Subspace Inference for Bayesian Deep Learning

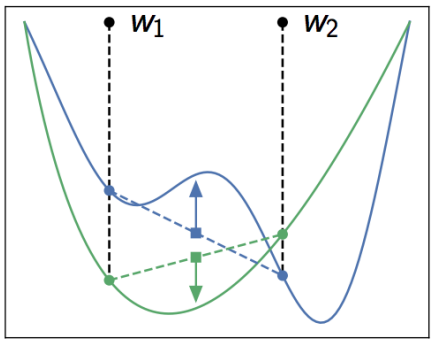
P. Izmailov, W. Maddox, P. Kirichenko, T. Garipov, D. Vetrov, A.G. Wilson

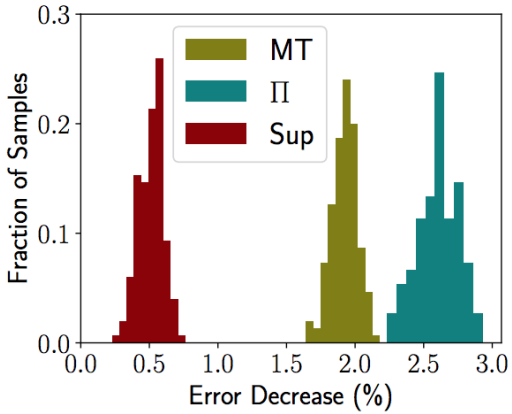
Semi-Supervised Learning

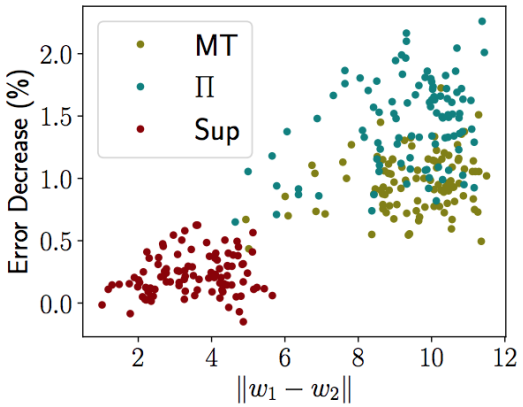
- Make label predictions using structure from both unlabelled and labelled training data.
- Can quantify recent advances in unsupervised learning.
- Crucial for reducing the dependency of deep learning on large labelled datasets.



Semi-Supervised Learning







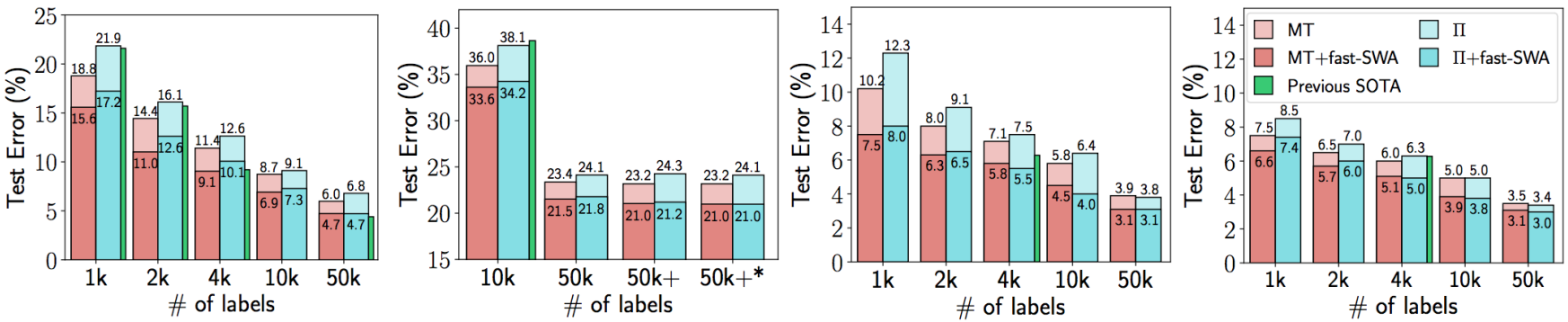
There Are Many Consistent Explanations of Unlabeled Data: Why You Should Average

B. Athiwaratkun, M. Finzi, P. Izmailov, A.G. Wilson

ICLR 2019

$$L(w_f) = \underbrace{\sum_{(x,y)\in\mathcal{D}_L} \ell_{\text{CE}}(w_f, x, y)}_{L_{\text{CE}}} + \lambda \underbrace{\sum_{x\in\mathcal{D}_L\cup\mathcal{D}_U} \ell_{\text{cons}}(w_f, x)}_{L_{\text{cons}}}$$

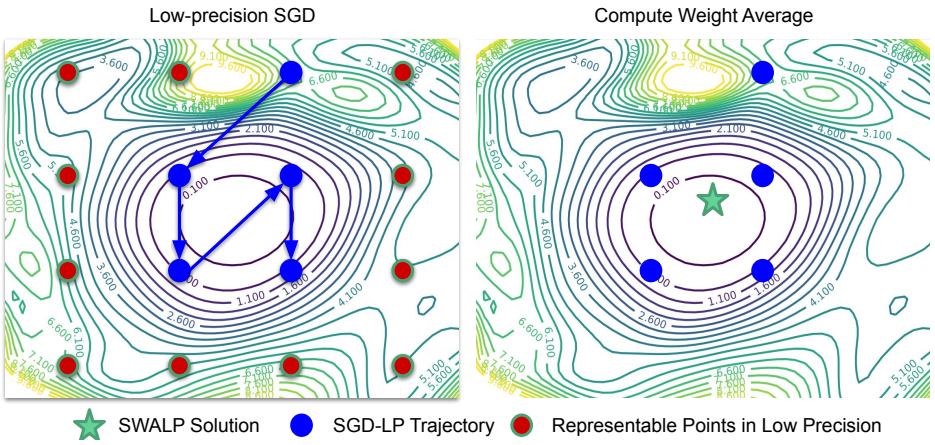
Semi-Supervised Learning

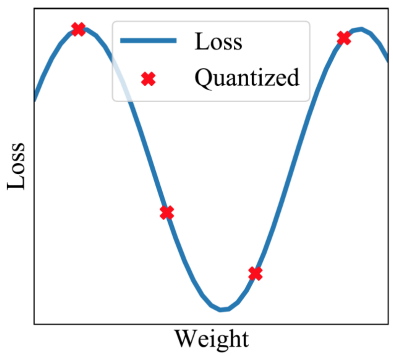


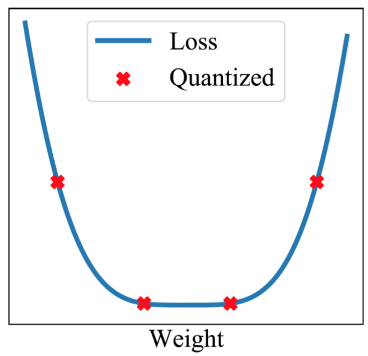
World record results on semi-supervised vision benchmarks

SWALP: Stochastic Weight Averaging in Low Precision Training

- End-to-end training entirely in low precision.
- Can outperform *full-precision* SGD even with all numbers quantized down to 8 bits, including gradient accumulators.
- Averaging combines weights that have been rounded up with those that have been rounded down.
- Quantizing in a flat region does not hurt loss.
- SWALP converges arbitrarily close to the optimal solution.
- Special relevance to new GPU architectures.







Conclusions

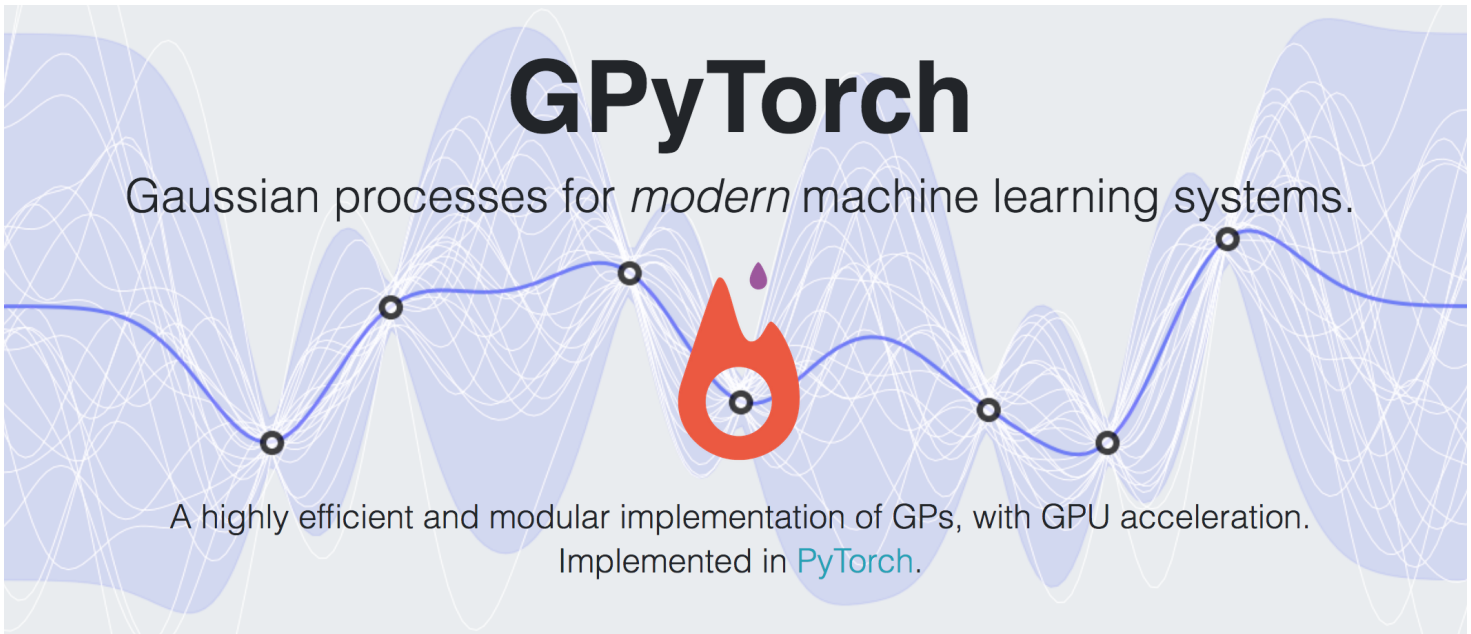
By considering the geometry of the loss surfaces, we can:

- Develop optimization procedures which provide better generalization, and good performance for low precision training.
- Develop scalable approaches to Bayesian deep learning, which both provide better point predictions, as well as uncertainty representation and calibration.

Code is available: `https://people.orie.cornell.edu/andrew`

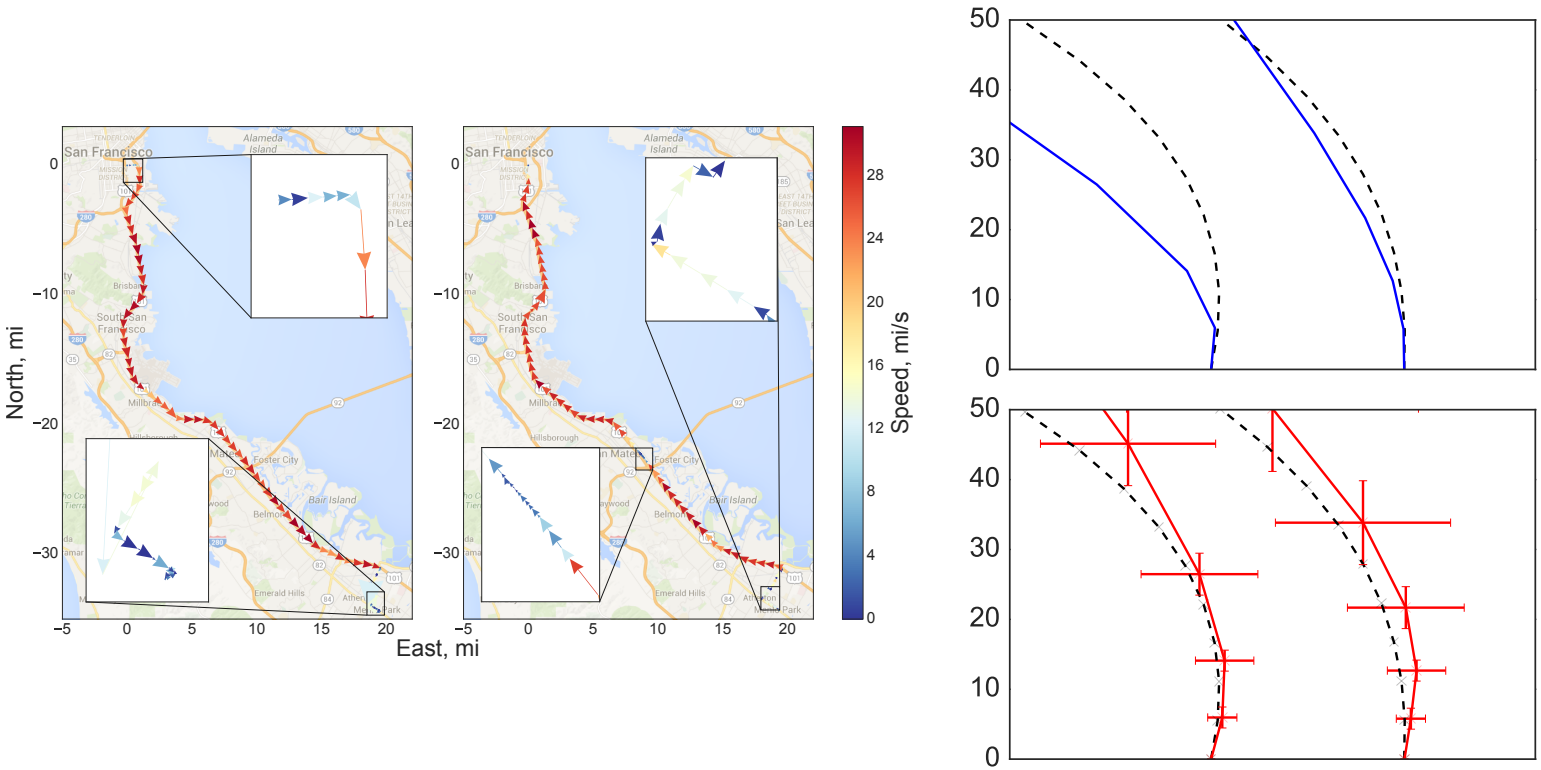
Scalable Gaussian Processes

- Run GPs on **millions** of points in **seconds**, vs. thousands of points in hours.
- Outperforms stand-alone deep neural networks by learning **deep kernels**.
- Approach accelerated by kernel approximations which admit fast matrix vector multiplies (Wilson and Nickisch, 2015).
- Harmonizes with GPU acceleration.
- $\mathcal{O}(n)$ training and $\mathcal{O}(1)$ testing (instead of $\mathcal{O}(n^3)$ training and $\mathcal{O}(n^2)$ testing).
- **Implemented in our new library GPyTorch**: `gpytorch.ai`



LSTM Kernels

- We derive kernels which have recurrent LSTM inductive biases, and apply to autonomous vehicles, where predictive uncertainty is critical.



Learning Scalable Deep Kernels with Recurrent Structure
M. Al-Shedivat, A. G. Wilson, Y. Saatchi, Z. Hu, E. P. Xing
Journal of Machine Learning Research (JMLR), 2017

GP-LSTM Predictive Distributions